PROCEEDINGS

VOLUME 2

INTEREX HP Users CONFERENCE AUGUST 20-23

MPE, MPE XL,
O.A.,
Networking,
Management

Sponsored by **INTEREX** The International Association of Hewlett Packard Computer Users

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

INTEREX

The International Association of Hewlett-Packard Computer Users

PROCEEDINGS

of the

1990 INTEREX HP Users Conference

MPE, MPE XL, O.A., Networking, Management

at

Boston, Massachusetts
August 20-23. 1990

Volume 2

Computer Museum

David W. Haberman, Editor

Integrated Image Applications

by

Malcolm Sharratt

Hewlett Packard Company
Pinewood Information Systems Division
Nine Mile Ride
Wokingham
Berkshire
England

Introduction

This paper will introduce the reader to the concepts of integrated Image Management Systems as embodied in Hewlett-Packard's Advanced Image Management System (HP AIMS). It will cover the make up both in hardware and software of such a system, deal with the tools supplied to create flexible, powerful user interfaces and outline the ways in which two of our customers are already using the product.

Many Image Management Systems sold today are targeted to meet a specific, insular need such as the records processing departments in large companies. This is not the way HP chooses to position its solution. Integration is key. The remainder of the paper will deal with ways in which we achieve integration both with HP systems and other vendors' proprietary systems. This will cover areas such as the use of PC based systems offering programmatic access to remote systems and Terminal Emulators.

This paper is targeted at Data Processing Managers, Systems Analysts and Programmers who have some knowledge of Microsoft Windows and a high level language, preferably 'C'. The paper does not assume that the reader has any previous knowledge of HP AIMS.

Image Processing Concepts

The base concept of Document Image Processing is the ability to take an electronic form of a paper document and allow the display, manipulation and, if necessary, printing of the image without the need to refer to the original document. With 95% of the world's information stored on paper, the attraction of such a concept can be seen at once.

After the initial need to remove the sheer volume of paper has been met and the cost savings associated with reduced office storage space requirements realised, other benefits emerge. Improved clerical throughput, better document tracking, enhanced responsiveness to customer queries and many more. But these benefits soon pale into insignificance when users realise that **Integration** with existing electronic information is what is really required.

This is why the HP AIMS concept is that of extending existing databases by allowing the effective inclusion of images. By basing our systems on industry standard operating systems, LANs and hardware, such integration is made possible.

We are seeing systems that today handle documents on paper, microfilm, microfiche and FAX format as well as data stored in unstructured formats such as PC files. In the future, we expect to see voice and video images, as well as colour. Other requirements we expect will be the alteration or "mark up" of images, the addition of handwritten comments and the efficient transmission of images to other geographic locations.

HP AIMS Hardware Components

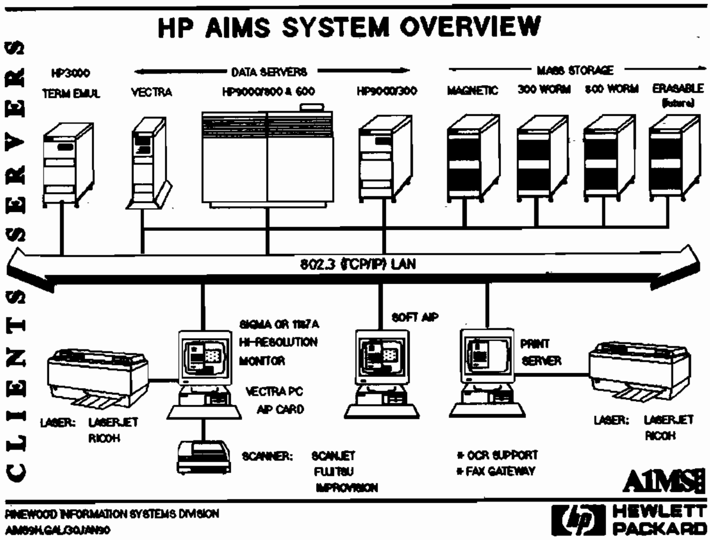


Figure 1

The HP Advanced Image Management System adheres to a classical Client/Server based architecture. The Server is any HP9000 Series 300 Workstation or Series 800 mini computer running the HP-UX operating system. As a future addition to the program, we will be providing support for a range of 80386 and 80486 PC servers running the Santa Cruz Operation UNIX operating system. These systems will be aimed at pilot schemes, low volume applications and initial system development. Attached to the server can be optical discs, currently using WORM technology and in the future using Magneto Optical (Erasable) technology, to support high volume applications and "On-Line" archive. The LAN conforms to 802.3 standards.

The client part of the architecture comprises an HP Vectra IBM PC/AT compatible 80286 or 80386 PC. The PC must have at least 4Mb EMS memory and in most cases an additional hardware board having between 2 and 6Mb memory plus dedicated hardware which performs compression/decompression, scaling and rotation. This is called an Advanced Image Processor board or AIP. A range of screens varying in resolution can be used, dependent on application. Scanning and printing are also controlled by interaction with the AIP board and OCR cards can be added as required, either on a per workstation basis or as dedicated servers.

Connections to remote mainframes can be made by standard serial cards or through gateways positioned on the LAN and handling many interactive sessions. The HP OfficeFax FAX server will also be able to be connected to the dataserver to allow the facsimile transmission of images.

HP AIMS Software Components

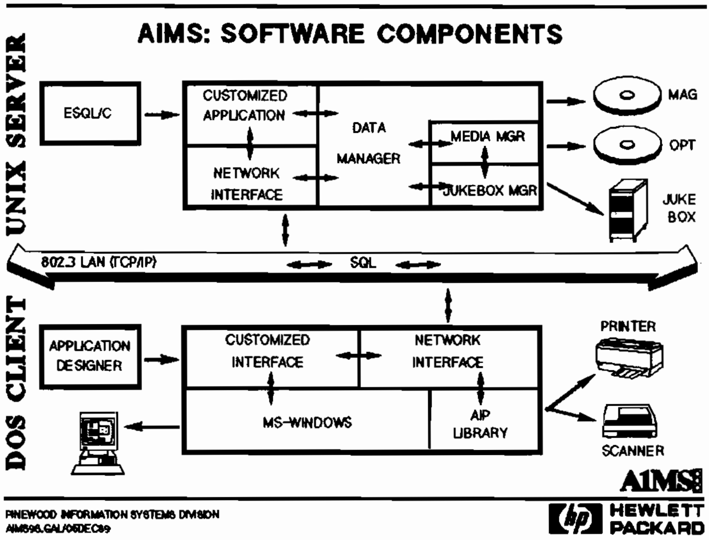


Figure 2

The HP AIMS software components are split into two distinct groups, those that reside on thc server and those that reside on the client PC's.

The heart of the dataserver software is a Relational Database Management System, known as the Datamanager. It has been enhanced to allow the handling of large unstructured data types up to 2 Gigabytes in size. These data types can be designated as images, text information or unspecified data types, useful for the storage of items such as PC files and programs. The database controls both magnetic and optical media as areas of raw disk outside the control of the UNIX file system. The Datamanager handles all i/o to the areas of magnetic disk. Storing information to optical disk requires that a pointer be written to the record index to identify the new media. The resolution of this pointer is performed by the Media Manager with control of multiple optical platter Jukeboxes enabled by the Jukebox Manager.

TCP/IP is used across the LAN to enable Berkley Socket connections to be established between Client and Server with extended standard query language (SQL) calls being used for database manipulation. The use of such high level SQL access to the database gives a more productive development environment as the application programmer has no need to be familiar with low level data communications protocols. As an added extra, an ESQL/C component can be purchased which allows the construction of programs on the data server which can manipulate the content of the image database.

The Client PC's split into two groups, those used for development which are described in the following section and those used in a production environment. The client PC's use MS-DOS and MSWindows to run programs which manipulate the AIP board and issue SQL requests to the Image server.

Development Environment

As the previous section stated, the development environment for HP AIMS is substantially different to the Production environment. The programs which run in a production environment are standard MSWindows executables which are known to be awkward and time consuming to construct. To overcome these difficulties, HP AIMS delivers a set of development tools which can increase programmer productivity for MSWindows development by 10-20 times! These tools consist of a 4GL and a User Interface Builder.

The User Interface Builder (UI Builder) allows the developer to graphically display the interface to his program by using an MSWindows program. Using a mouse, he can position, size and annotate forms, menus, buttons and boxes at will, giving them each a unique reference which is later used in the 4GL code he writes to manipulate the user interface. The end result of this program is to produce the RC file familiar to all MSWindows programmers. The major advantages here are the speed and flexibility with which a designer can develop an interface, also the ease with which the interface can be modified if required.

A comprehensive 4GL with compiler is the other major development component. This uses the references generated in the UI Builder to manipulate and control the user interface. Constructs such as Form Handlers and Menu Handlers, control Forms and Menu Bars. IF, WHILE, FOR and CASE statements control program logic and if necessary, EVENT clauses control the intercept of any MSWindows messages. Functions are made available to control the manipulation of images that reside on the AIP & SQL statements are written in line to control the retrieval of data from the server.

Examples of Use

Having given a very brief overview of the Concepts and Components that make up HP AIMS, it may help to give examples of how two of our customers are using the product.

The first is a major European bank who are currently piloting what will be a system far in excess of 100 users to process company and personal loans. They require basic workflow capabilities to route the scanned forms to a predefined set of users, the ability to authorise the forms at different stages and add comments on an ad hoc basis. Some access is required by remote workstations over an X.25 link and casual viewing on workstations with non high resolution monitors. Also key to their requirement is the need to integrate these images with databases which already exist on HP3000's in their organisation, a technique that will be covered in more detail later.

Another example is that of a medical litigation service which is currently at the pilot stage. They have the need for up to 50 personnel to maintain and access complex files concerning insurance claims. These files contain documents of many types giving expert opinions on cases and historical performance as well as current estimates of liability. The physical make-up of this system is worth mentioning as the dataserver with 5-10 users is based in one city with the remaining users being 200 miles remote from this and connected via a 2 Megabit per second communications link. As each specialist report commissioned costs money, effective use of these is required. For this reason, the customer requires to search on the text content of the report. To achieve this they are using a proprietary Full Text searching engine and OCR techniques. The access to the Full Text Engine is to be seamlessly written into their HP AIMS applications. They too use a limited form of workflow, using a very simple implementation of electronic mail. In the future, they intend to install smaller systems in other countries which will need to receive and transmit data from the main dataserver on an infrequent basis whilst performing the role of satellite systems for local Image Processing.

Integration

The rest of this paper is about the integration of HP AIMS with existing data processing systems. The easiest way to understand this type of requirement is to consider a simple example.

Company X has a Manufacturing Management System based on an Image/3000 database with a User Interface written using V/3000. The Manufacturing System holds information on parts (amongst other things), with the PART-MASTER dataset holding information of the form:-

```
PART NUMBER         INTEGER
PART DESCRIPTION    X(30)
.
.
.
DRAWING NUMBER      INTEGER
.
.
```

One of the users of the system is the Goods Inwards Inspection Clerk. His job is to check that externally manufactured parts meet certain tolerances and to do this he has to make a series of measurements of the component whilst referring to the A4 drawing of the part. He has a terminal on his desk that he uses to update the Manufacturing Management database on the HP3000. As there are a large number of parts in use by the company, the formal drawings are only kept on microfilm to save space. When a new part arrives for him to check he has to follow this procedure:-

* Call up the Part-Master Record for the part in order to find the Drawing Number
* Note down the Drawing Number
* Physically move to the microfilm index
* Search for the Drawing
* Make a copy of the drawing (potentially poor quality)
* Take the copy back to his workbench
* Perform the required checks on a representative sample of the parts
* Call up the Stock Record for that part to update the quantity of the part accepted
* Throw away the copy of the part drawing

Installing HP AIMS to hold all of the Engineering Drawings in place of the microfilm unit in a non-integrated way would improve efficiency. Replace the terminal with an HP AIMS workstation running both an HP AIMS application to provide image access and HP AdvanceLink for Windows to provide Terminal Emulation and the steps become as follows:-

* Call up the Part-Master Record for the part in order to find the Drawing Number
* Note down the Drawing Number
* Enter the number into an HP AIMS application causing retrieval of the previously scanned drawing
* Perform the required checks on a representative sample of the parts
* Call up the Stock Record for that part to update the quantity of the part accepted

This may be adequate for some applications, but most will require that further step towards truly integrated applications that is given by the entry of one key value causing the automatic retrieval of both database information and image. This can be done in two ways, either with or without using a Terminal Emulator.

Seamless Applications

One way is to produce a seamless application in which the user does not see that data is being retrieved from two sources. Each major manufacturer has a product that will allow programmatic communications between PC and remote host. Hewlett-Packard's offering is HP Cooperative Services.

This is a Programmer's tool that allows the creation of coprocessing applications that span the PC and HP3000 environments. Added to HP AIMS which is in effect a programmer's tool that allows the creation of coprocessing applications that span PC and HP9000, the power of the combination can be seen, integrated databases on both HP3000 and HP9000 joined at the PC by one application.

Put simply, this means that MSWindows programs can be written that access databases and files on the HP3000 via pseudo IMAGE/3000 intrinsic calls and databases on the HP9000 via SQL calls. The user interface of these programs can hide the location of the data extracted from the different sources, making the user believe that the location of the information is one and the same.

Cooperative Services resides on both the PC and HP3000 Server. On the PC, Library calls are made from the application program which mimic Image/3000 intrinsic calls. These are passed through a Requestor (a Terminate and Stay Resident program preloaded into PC memory) where they are translated into requests for dispatch to either OfficeShare (LAN connections) or the Basic Serial Library (Serial Connections). Replies to these requests follow the reverse order. The host communications structure differs between LAN and Serial connections. For Serial connections, a PC Server handles the initial requests which are translated and passed to the Host Server where they are turned into the necessary File access calls. For OfficeShare connections, the PC Server is replaced by AdvanceNet. (See Figure 3).

One other area worth noting is that data is held in different structures on the PC to those on the HP3000. Due to this it is necessary to perform conversions of some data items prior to transmission and on receipt. This job is performed by the PC Programmer using routines supplied by HP Cooperative Services.

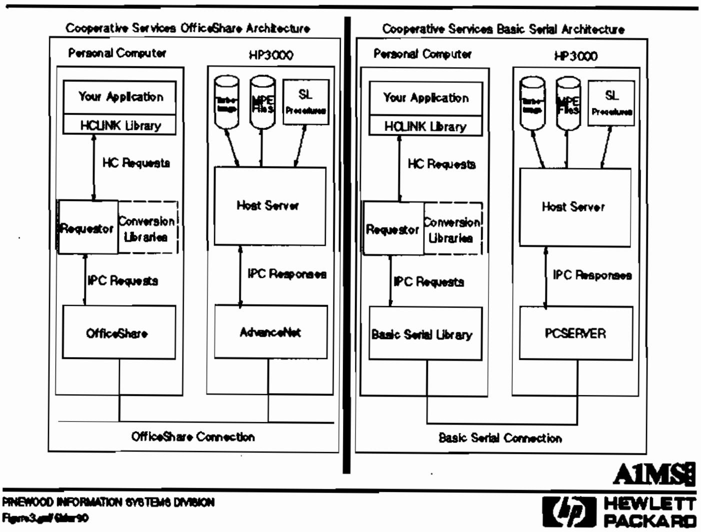


Figure 3

How then do we go about constructing such a program using the Development Tools available to the HP AIMS Developer. The actual process is quite simple. A User Interface is developed using the UI Builder, defining areas to be used to hold the data from both sources. The 4GL is then written to control this Interface. As well as providing UI control, it must also access the data from both sources. To access the data on the HP AIMS dataserver is simplistic, using SQL statements for retrieval and DISPLAY statements to output. The access of the data on the HP3000 is more complex.

The 4GL has a statement (CALL) which allows the programmer to execute functions out of the main line code. These functions can either be written in 4GL or in the 'C' programming language. The calls to 'C' functions are the mechanism used to access the HP Cooperative Services functionality. However, just CALLing the HP Cooperative Services functions from the 4GL application is not enough and to explain why, it is necessary to take a look at the 4GL calling mechanism.

The calling mechanism utilises a stack which can be accessed by both 4GL programs and 'C' functions. During every use of the CALL statement in the 4GL, the values of the parameters passed to the function are pushed on to the stack. The CALLed function then retrieves them from the stack and manipulates them. If any of the modified values are to be passed back to the CALLing function, they must again be placed on the stack. 4GL functions that are called take care of the unloading and loading of the stack automatically but foreign functions written in 'C' have to provide these actions themselves.

To explain more fully, we will examine the call to the DBGET function of HP Cooperative Services from a 4GL program. The initial call from the 4GL program is shown in Figure 4.

When this is passed through the 4GL compiler, it generates the code shown in Figure 5. This is standard 'C' code although it may not seem so at first glance. The CALL statement in the 4GL has been parsed and the parameters to be passed to c_dbget (those inside the brackets) have been loaded on to the stack using the **pushshort** and **pushquote** functions. These functions are contained in a library supplied with the HP AIMS Development Environment. Once all the parameters are loaded, the function c_dbget is called with one parameter, the number of parameters it will find on the stack.

On return, the number of parameters being passed back to the calling function is checked and if correct they are unloaded from the stack. The unload procedure uses the **ld_short,ld_long** and **ld_quote** Library functions.

The implications that this mechanism has on the way in which the CALLed 'C' function needs to be written can be seen in the function definition shown in Figure 6. The **pop** library functions are used by the application programmer to ensure that the passed parameters are removed from the stack in the reverse order. The **ret** library functions are used to ensure that the correct values are returned to the calling function to satisfy the variable referenced in the RETURN clause.

Repeating this for all required HP Cooperative Services library calls enables the user to write simple 4GL programs which access both HP3000 and HP9000 databases simultaneously. Needless to say there is a negative point to this procedure. The customer may well have invested significant human and financial resources into developing a secure and logical interface to their existing databases. To be required to reinvest time and money to replicate certain parts of the interface, even with the benefit given by a truly integrated application may not be viable. For these reasons, we are also targeting integrated Terminal Emulator solutions.

The above example only considered access to HP3000 systems. Using an alternative to HP Cooperative Services such as an implementation of LU6.2 could achieve access to IBM mainframes. The process to integrate such a system would be identical within the 4GL, with the programmer needing to cocoon the remote calls to handle the parameter passing in a correct manner.

Initial Call to c_dbget from 4GL Application

```
CALL c_dbget(gv_3000_base_id,gv_3000_base,lv_dset,gv_3000_mode,lv_list,
             lv_buffer_len,lv_arg_len,lv_target,lv_arg_type)
             RETURNING
             gv_host_status,lv_buffer, gv_3000_status.cond_word,
             gv_3000_status.word2,gv_3000_status.word3_4,
             gv_3000_status.word5_6,gv_3000_status.word7_8,
             gv_3000_status.word9_10
```

Figure 4

4GL Compiled Version of Call to c_dbget

```
{
int fgl_svstk,fgl_nret;
fgl_svstk = mark_stack();
pushshort(gv_3000_base_id);
pushquote(gv_3000_base,28);
pushquote(lv_dset,16);
pushshort(gv_3000_mode);
pushquote(lv_list,256);
pushshort(lv_buffer_len);
pushshort(lv_arg_len);
pushshort(lv_target);
pushshort(lv_arg_type);
fgl_nret = c_dbget(9);
rest_stack(fgl_svstk);
if(fgl_nret!= 8)
        status = -4503;
else
{
ld_short(0,&gv_host_status);
ld_quote(1,lv_buffer,257);
ld_short(2,&gv_3000_status.cond_word);
ld_short(3,&gv_3000_status.word2);
ld_long(4,&gv_3000_status.word3_4);
ld_long(5,&gv_3000_status.word5_6);
ld_long(6,&gv_3000_status.word7_8);
ld_long(7,&gv_3000_status.word9_10);
}
if(status < 0)
        AD_CallErrWarnFunc1(curr_file_name,91,rep_error);
}
```

Figure 5

Function c_dbget

```
c_dbget (nargs)
int nargs;/*Number of arguments on the stack*/
{ union BOVERLAY {int base_id;/*DataBase id set by DBOPEN*/
          char base_name[28];/*DataBase Name*/}base;
 char dset[16];/*DatasetId*/
 int mode;/*Must be set to one through 8*/
 char list[256];/*List of data items - could be up to 4335 chars*/
 struct {int length;
   union {char body[512];
        int ibody;} x;
   }arg;
 struct DBSTATUS {int cond_word;
          int word2;
          long word3_4;
          long word5_6;
          long word7_8;
          long word9_10;
          }c_status;
 struct BUFFERS buff;
 int arg_type;/*1=Numeric,2=Text*/
 if(nargs==9) /*Check that the correct number of arguments have been passed*/
 {  popint(&arg_type);
        if (arg_type==2)
        {popquote(arg.x.body,512);/*Textual search value*/}
        else/*Numeric*/
        {popint(&arg.x.ibody);}
        popint(&arg.length);/No of chars in arg.body*/
        popint(&buff.length);/*Length of buffer*/
        popquote(list,256);/*Data item identifiers*/
        popint(&mode);/*Must be between 1 and 8*/
        popquote(dset,16);/*DatasetId*/
        popquote(base.base_name,28);/*Base Name*/
        popint(&base.base_id);/*Base Id*/
        /*Call the function*/
        hcstatus = dbget(&base,&dset,&mode,&c_status,&list,&buff,&arg);}
 else
        {hcstatus=999;} /*Set hcstatus to unique value to signify invalid no. of   parms*/
 retint (hcstatus); /*Put hcstatus on stack ready for the return*/
 retquote(buff.body); /*Put buff.body on the stack for return*/
 retint (c_status.cond_word); /*Put elements of c_status array on stack for   return*/
   retint (c_status.word2);
 retlong(c_status.word3_4);
 retlong(c_status.word5_6);
 retlong(c_status.word7_8);
 retlong(c_status.word9_10);
 return(8); /*Return from the function giving the number of return arguments*/
}/*End of Function c_dbget */
```

Figure 6

Terminal Emulation

Using a Terminal Emulator to provide the integration with remote hosts enables the user to continue with his standard user interface whilst displaying other information, including images, in a separate window. To do this requires that the Terminal Emulator and the HP AIMS application both communicate in a way that each other can understand. The logical way to achieve this inside MSWindows is to use Dynamic Data Exchange (DDE).

For those readers not familiar with the workings of DDE, I shall cover the basics here. Some confusion may arise over the use of Client/Server terminology in this section. I am using MSWindows terminology in which both Client and Server applications run on the same PC, with the Server satisfying the Client's requests for data. MSWindows is a multitasking environment in which different applications or multiple copies (instances) of the same application can run simultaneously. In order to coordinate the execution of these tasks, MSWindows uses a message system where each message notifies an application of an event of interest which should prompt a specific action. One series of events that are governed by a select set of messages are those concerned with inter application communication (DDE). In a DDE conversation a **server** application supplies data to one or more **client** applications. it does this after the **client** has established a **conversation** with it. The **client** can ask the **server** for a one time transmission of data (a **request**) or a continuous transfer of data whenever it changes (an **advise**).

In the context of our example, the Clerk would start two applications on his workstation, the **server** application (the Terminal Emulator) and the **client application** (the HP AIMS application). The Terminal Emulator would be such as AdvanceLink for Windows and would be executing a script file which would be monitoring the transmissions from the HP3000, waiting for a particular event to occur (like data entered in a particular screen location or a certain character string transmitted from the mainframe). The client would be a standard HP AIMS 4GL application, enhanced to trap DDE messages as they were transmitted from the **server**. The Clerk would enter a part number in his Terminal Emulation window which would cause the retrieval of the Part Master data. This would then be transmitted to the HP AIMS application, causing the automatic retrieval of the A4 drawing of the part. The likely flow of information in such a case is as follows:-

- *Clerk starts HP AIMS application.
- *HP AIMS Application automatically starts Terminal Emulator (TE) and script.
- *HP AIMS Application initiates **conversation** with TE.
- *TE replies that it can understand.
- *Clerk enters data into TE and retrieves data.
- *TE script recognises change of drawing number.
- *TE advises HP AIMS Application of change of data.
- *HP AIMS application issues Select statement, retrieves record and displays it.

On the negative side, MSWindows based Terminal Emulators that support DDE and true HP Terminal Emulation are at present non-existent. This will not be solved until versions of AdvanceLink for Windows are made available in the late 1990, early 1991 time frame.

Providing that a Terminal Emulator can be found that will communicate with a remote host through a gateway connected across a 802.3 LAN using OfficeShare (potentially with NETBIOS) or via a serial interface of some sort (ie IRMA card), communications to other vendors' hosts can be established

Conclusion

This paper gave an outline of the concepts behind the business of Image Processing. It also covered the make-up of Hewlett-Packard's answer to the corporate need in both hardware and software terms.

It then examined the way in which truly integrated image processing systems can be achieved, either using dedicated program to program communications or the more flexible approach of Terminal Emulators.

I believe that this has shown the importance that Hewlett-Packard is placing on the integration of HP AIMS into existing data processing departments and the way in which we are approaching the opportunities raised by this approach.

Designing Multivendor LANs That Work

Robert S. Yori
Hewlett-Packard Co.
3301 Royal Lane
Irving, Texas 75063

The toughest challenge facing Local Area Network (LAN) designers today is determining how to integrate and managing corporate networks composed of PCs, minis, and mainframes - all potentially from different vendors.

Common questions are:

A. How can diverse local area network technologies be molded into one network supporting common software applications, while using different communication protocols?

B. How can the various hardware devices be managed from one central control center?

C. What are the problems that can be encountered?

These issues are addressed in this article. Initially, the focus is on the technology currently available for designing LANs with hardware and software from various manufacturers. An example of this technology will be reviewed - Hewlett-Packard's Novell Gateway. Next, tips for designing these multivendor networks will be discussed. The final section outlines emerging developments that will make it possible to better manage multivendor networks.

Before discussing any details, let's review the example of how these LANs can be constructed. Figure 1 pictorially represents this.

Here we have a Hewlett-Packard OfficeShare PC network, composed of HP 3000 minicomputers and IBM-compatible PCs. OfficeShare is a collection of products allowing the HP 3000 to function as a server on a LAN. Because of the requirement for a particular software package, another department within the company has a Novell NetWare LAN. The Novell users would like to run applications on the HP 3000 computers and transfer files between the PCs and the HP 3000. Also, software applications on the PCs must communicate in a program-to-program environment with applications on the HP3000.

In order for the Novell PC users to communicate as remote terminals over the Novell NetWare LAN to the HP OfficeShare LAN, the following is required:

A. The Novell PCs in the network must run a terminal emulation software program such as HP AdvanceLink. This would be in lieu of other Novell user applications.

B. Physically connect the two LANs with a PC gateway running a protocol-translation program. This is required because the NetWare and OfficeShare LANs use different network communication protocols (figure 4). This gateway PC must also be configured with both Novell and HP LAN cards.

In order for the PC users to transfer files to the HP 3000, the terminal emulation software HP AdvanceLink would be used. This file transfer application must be running on the Novell PC and the HP 3000. A PC gateway would still be necessary. Figure 1 pictorially represents this scenario.

In general, other terminal emulation software packages could be used instead of HP AdvanceLink. For example, the defacto industry standards TELNET (TELetypewriter NETwork protocol) could be used for terminal emulation, and FTP (File Transfer Protocol) for file transfers. As with HP AdvanceLink, both TELNET and FTP would have to be installed on the PC and the HP3000.

The program-to-program issues will be reviewed later when the technology used to design the HP Novell Gateway is discussed.

One comment concerning TELNET. TELNET does not support all terminal protocols, such as DEC VT-240. For full-screen HP 3000 block mode support, special versions of TELNET are available from HP. Both TELNET and FTP are part of a set of software services called ARPA (Advanced Research Project Association) - an organization within the U.S. Department of Defense.

TECHNOLOGY

The software technology making it possible to connect different LANs is the communications model developed by the International Standards Organization (ISO). The ISO is an organization within the United Nations with the charter to develop and promote worldwide communication standards.

The model is called the Open System Interconnect (OSI) model. It is a software methodology providing for communications between computer systems of hardware manufacturers who implement the model - a model that is independent of the computer's operating system. Figure 2 shows this 7 layer OSI model. Within the definition of the 7-layer OSI model:

A. Any given layer communicates or passes data to adjacent layers (figure 2)

B. The same layers on different computers in the network communicate

For example, layer 2, the Data Link layer, sends and receives data to and from layers 1 and 3. And, layer 2 on one computer communicates with layer 2 on another computer. One example of this is the error correction algorithm performed by layer 2 - the CRC (Cyclical Redundancy Check). Layer 2 on the sending computer receives confirmation from layer 2 on the receiving computer that the packet was either received correctly, or retransmission is required.

Within the structure of the OSI model, the task of moving information from one user's computer, across a communication network, to another computer is subdivided into 7 tasks or layers. At each layer, the OSI model defines the specific activities performed, plus the rules for communicating with adjacent layers. Each layer is a separate set of rules - in effect a separate protocol - making it possible for different computer systems, PCs, or desk-top work stations to communicate, regardless of the operating system used by each computer. The 7 layers together are called a **protocol stack.**

Given that one particular layer of the OSI model interacts with only adjacent layers, the theory behind the OSI model states that layers are "removable". This feature makes possible, for example, the substitution of software and hardware running coax LAN communication (layers 1 and 2) to be replaced by that running LAN communication over unshielded twisted pair wiring - without affecting layers 3 through 7. Thus, communication protocols can be easily adapted for various vendor and communication environments.

Even though all layers are "removable" within the OSI model, certain layers work more closely with each other. These closely coupled layers are:

Layers 6-7

Layers 3-4

Layers 1-2

Figure 3 shows this relationship between the layers that are closely related. One example of this is the interaction between layer 3 - the network layer, and layer 4 - the transport layer. Layer 3 routes packets through the network. Layer 4 ensures that all packets have arrived, and are sent to the upper layers in the correct sequence.

Consider the situation where the user data being sent through the network will not fit into one packet. There is a possibility that the packets may arrive at the receiving end in a different order. This can happen when the retransmission of packets is required - based upon layer 2 CRC checking, or when alternate routing paths are used - based on layer 3 routing algorithms. Layer 3 will pass the packets to layer 4 in whatever order they are received. Layer 4 then sequences them correctly before passing the user data to layer 5.

Layer 5 is a pivotal layer, connecting the application interface layers 6 & 7, with the transport and network layers 3 & 4 (see figure 3). It is a set of application program interfaces (APIs) or subroutines that are programatically callable from the user application.

Examples of protocols used at layer 5 include:

A. NetBIOS - developed by Sytek for IBM

B. NetIPC - developed by Hewlett-Packard

C. BSD Sockets - a Berkeley extension to UNIX.

In all instances, layer 6 must know which protocol is being used at layer 5. If an application is to run on protocol stacks supporting either NetBIOS or NetIPC, then there must either be two versions of the application, or both interfaces must be designed into the same version.

Also, situations occur where LANs using different protocol stacks must communicate. For example, a PC on an HP Officeshare network may need to share files with a PC on an IBM token ring network. This may be necessary when users select software packages that run only on one particular PC LAN. In this case, a gateway can be used to translate the 7 layers of one OSI protocol stack into the 7 layers of another OSI protocol stack. This gateway can be a PC running special software, or a hardware device built specifically to handle the translation. In either case, the gateway must be configured with LAN interface cards for both networks. The software running on the gateway must translate one LAN's protocol stack into another LAN's protocol stack. In effect, the gateway supports dual protocol stacks. Figure 1 pictorially represents the functioning of the gateway referred to in the first example.

The feature of **layer removability** within the OSI model makes the construction and maintenance of a multivendor LAN running a common user application possible! This feature gives the network designer several options:

A. Match all 7 layers of the protocol stacks on all computers. This may be the most difficult to do, especially in a large network, since no single computer manufacturer makes all the software and hardware that may be needed by all users.

B. Match layers 3 through 7, and use bridges to translate the layer 1 and 2 protocols. This can be done in situations where network software is supported on different physical networks, such as with Novell.

C. Design the network so that different LANs, running on different protocol stacks, communicate by way of gateways.

The next section details this last option, showing how gateways make communication between different LANs possible.

EXAMPLE

Let's take a closer look at the design of the HP Novell Gateway product. As discussed earlier, this is a PC-based gateway, translating a Novell protocol stack to an HP Officeshare protocol stack. This allows HP Officeshare software to run on the PCs with a Novell protocol stack, while preserving all the features of the HP Officeshare PC software. This feature set includes:

Terminal emulation and file transfer via AdvanceLink

Program-to-program communication via NetIPC

"Cooperative Services" allowing the HP3000 to function as a PC server on the LAN

Figure 4 shows a comparison between a Novell protocol stack, an HP Officeshare protocol stack, and the OSI model. Here, the Novell and HP stacks are different at each of the 7 layers - the Novell stack is running on an 802.5 IBM token ring (TRN) physical network. Novell networks operate on physical networks other than TRN, such as Ethernet, 802.3, and ArcNet. The HP Novell Gateway supports all of these interfaces. In this discussion, the TRN interface will be used.

Figure 5 shows the relationships between the HP and Novell protocol stacks as packets move from the PC, through the gateway, to the HP 3000. The common portion of the packet that is transmitted through the network is shaded.

User PC:

Here, the upper 5 layers of the HP Officeshare protocol stack (layers 7 - 3) are passed to NetBios. These upper 5 layers are passed, via the NetBios SEND command, as a block of user data. At this point, NetBios then passes this "data" to the Novell transport and network layers (layers 4 and 3), and then onto the network through the TRN card.

HP Novell Gateway:

The gateway receives a packet from the user PC through its TRN card. This packet is then sent through the Novell protocol stack within the gateway. At layer 5, NetBios passes the block of "data", which originated in the user PC, to the HP gateway software. This special software links the "data" into the HP Officeshare protocol stack on top of layer 2. This is possible since the "data" from the user PC is really the upper 5 layers of the Officeshare stack. The packet of information is then sent through the 802.3 card in the gateway to the HP 3000.

HP 3000:

The HP 3000 receives the packet from the gateway, and sends it through the Officeshare protocol stack. At this point, the incoming packet looks like any other 802.3 Officeshare packet. The HP Novell Gateway eliminates the need for the HP 3000 to know anything about the Novell LAN.

This scenario describing the flow from the user PC, through the gateway, to the HP 3000 is reversed when the HP 3000 sends packets back to the PC user.

This design has several key advantages:

A. HP Officeshare software that currently runs on 802.3 LANs will run through the Novell Gateway without modification and without loss of functionality. This is a major savings for HP, software suppliers, and users.

B. By bundling the upper 5 layers of the Officeshare stack on the user PC on top of the Novell stack, the gateway performance is increased. The need for the gateway to add or remove the layer 5, 4, and 3 network information to the PC application data is eliminated.

C. The ability for PC applications to communicate with HP 3000 applications in a program-to-program environment is preserved. This is because the layer 5 NetIPC APIs are part of the common data sent with all the packets in the network (figure 5).

As mentioned before, this implementation is independent of the physical LAN card used by the Novell network software. The HP gateway software in the PC gateway only communicates with NetBios. The Novell network software handles this interface to all the network cards supported by Novell.

DESIGN SUGGESTIONS

When software and hardware components are supplied by various companies. it may be difficult to ensure the interoperability of these components. To avoid these problems, choose network components that conform to international standards, specifically the OSI communications model. Testing is still required though. This is one of the charters of the Corporation of Open Systems (COS) - to test selected software for conformance to the OSI standards. Software not already tested for interoperability must be tested by the designer of the network.

When designing networks, one common misconception is that Ethernet and 802.3 are the same, since many times the terms are used interchangeably. In reality, the 802.3 standard grew out of the Ethernet definitions defined by Xerox Corp. Both of these protocols use the same physical cabling. But, the format of the data packets sent over the communication network is not the same. The difference is in the format of the data packet after the destination and source addresses. The end result is that systems talking Ethernet and 802.3 can use the same cabling, but cannot talk to each other. Packet conversion is required for Ethernet and 802.3 systems to communicate. Two companies that make hardware and software products that perform this conversion are The Wollongong Group (Palo Alto, Calif.) and Cisco Systems (Menlo Park, Calif.). For a more detailed discussion concerning Ethernet and 802.3 differences, refer to the September, 1988 issue of "Interact" magazine - page 58. "Interact" is an HP installed-base publication.

Also, when designing a LAN, assume that at some point the network will need to accommodate different communication protocols, along with various application packages. These abilities will be inherent within the network if protocols and applications that conform to the OSI standards are specified. As the designer, you will then be able to take advantage of the concept of **layer removability** to install applications throughout the network, using whatever communication protocol is appropriate within a department or work group.

Finally, the best way to control LAN performance is to isolate a department's data traffic from the main LAN backbone by using a device called a "router", or "IP router". A router manages the flow of packets through the network by reading the layer 3 IP address, hence the term IP router.

If the address is for a device on the departmental LAN, the packet never gets sent onto the backbone, thereby isolating the departmental traffic and increasing the performance of the entire network. A gateway can perform the same functions as a router, but the gateway reads all 7 layers of a protocol stack, where a router reads only the first 3 layers.

One additional note on gateways and routers. Since gateways and routers can perform the same tasks at layers 1 through 3, the question is which device to use to link local area networks. Routers are hardware devices designed for a particular task, many times with the layer 1 through 3 software implemented in read-only memory (ROM). Gateways are usually PCs or minicomputers which run software that translates all 7-layers of a protocol stack. Therefore, gateways will tend to be more expensive than routers. Plus, gateways are slower, since their software applications are not ROM-based, and they must translate 7 layers of protocol.

NETWORK MANAGEMENT

Network management is one aspect of LAN design that is overlooked more often than not during the network planning phase. When planning a multivendor network, the ability to effectively manage the network and its components should be designed into the network before any vendor is selected.

Network management implies different capabilities to different designers. In its basic form, network management provides for the detection of component failures in the network. The emerging concepts of network management provide for the management of the PCs and computer systems on the network. This includes monitoring the performance of the network and the computers.

The emerging OSI standards define software procedures for managing networks from the perspective of:

A. Determining sources of failure

B. Tracking network and system performance

C. Device inventory management

D. Security

On the TCP/IP platform (layers 3 and 4), one defacto standard is Simple Network Management Protocol (SNMP). For the OSI platform, the network management scheme emerging is Common Management Interface Protocol (CMIP). One of the advantages of CMIP over SNMP is that CMIP provides for network security.

Communication and hardware vendors committed to implementing OSI communications protocols will also be implementing CMIP. These vendors today run on TCP/IP protocol stacks - many with SNMP already supported. The market trend is that vendors will migrate to OSI protocol stacks from TCP/IP protocol stacks. So, by implementing a TCP/IP network now using SNMP as the network management protocol, you will be able to migrate to OSI protocols using CMIP.

Some OSI communication protocols already on the market are:

A. TP0, TP2, and TP4 - TCP transport layer replacements

B. MAP - Manufacturer's Automation Protocol

C. X.400 - electronic mail interface standard

Note that "OSI communication protocols" are specific protocols (TP4, MAP, X.400, etc.) that conform to the OSI 7-layer communications model. For example, TP4 is an OSI protocol running at layer 4, where X.400 and MAP are layer 7 protocols.

When designing a network on the TCP/IP platform, specify that the network management software not only use SNMP now, but that the vendor will migrate the software to the CMIP standards.

The HP Openview Network Management software today implements SNMP on a TCP/IP protocol stack. HP is one vendor committed to migrating to OSI from TCP/IP.

CONCLUSIONS

We have discussed techniques that can be used to create an environment where multivendor LANs not only communicate, but support common applications. One technique discussed was to use gateways to translate different network protocols.

One major advantage of designing a multivendor network based on the OSI model is that common user software packages can be used on all the computers and PCs in the network. HP AdvanceLink, as discussed in this article, is just one such software package.

Common user interfaces can also be used across the different platforms - such as HP NewWave. This use of a common application interface on all computers will ensure that users can operate any of the computers in the network, regardless of their operating system. This common user interface shields the user from the particular operating system. The network designer is then free to choose the appropriate technology best suited for a particular segment of the network. The network can truly be designed for both the convenience of the user, and the flexibility and performance needed by the network designer.

By using LAN hardware and software components based on the OSI model, software applications can be purchased from various suppliers. The purchase criteria should depend on whether the application can communicate with the layer 5 protocol being used in the network. This gives users not only a larger choice of software products, but provides the network designer and corporation economic flexibility. Companies now have the ability to purchase applications for the network, and network hardware components, from vendors other than the original supplier.

Another advantage of specifying conformance to OSI standards is that, in the near future, network management will be a part of that standard (CMIP). This will allow centralized management of a multivendor network. The network manager will be able to determine the operational state of all components, including their performance.

The OSI standards provide economies of scale for the designer, the user, and the corporation. As the international standards evolve, the network can also evolve. Adherence to international standards ensures that the network can be maintained, supported, and enhanced for many years - thereby protecting corporate investments.

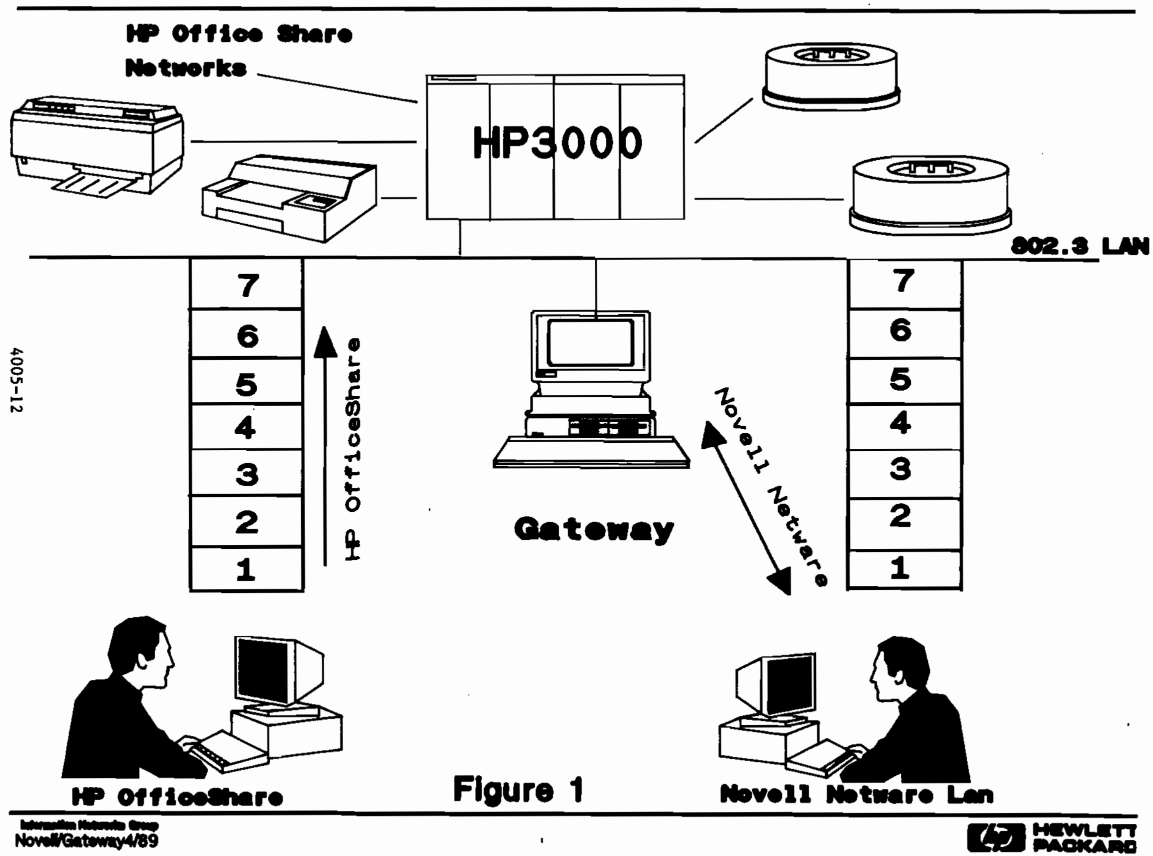

HP Office Share
Networks
HP3000
802.3 LAN
7
6
5
4
3
2
1
HP OfficeShare
Gateway
Novell Netware
HP OfficeShare
Novell Netware Lan
Information Networks Group
Novell/Gateway4/89
HEWLETT PACKARD

Figure 1

OSI 7-LAYER MODEL

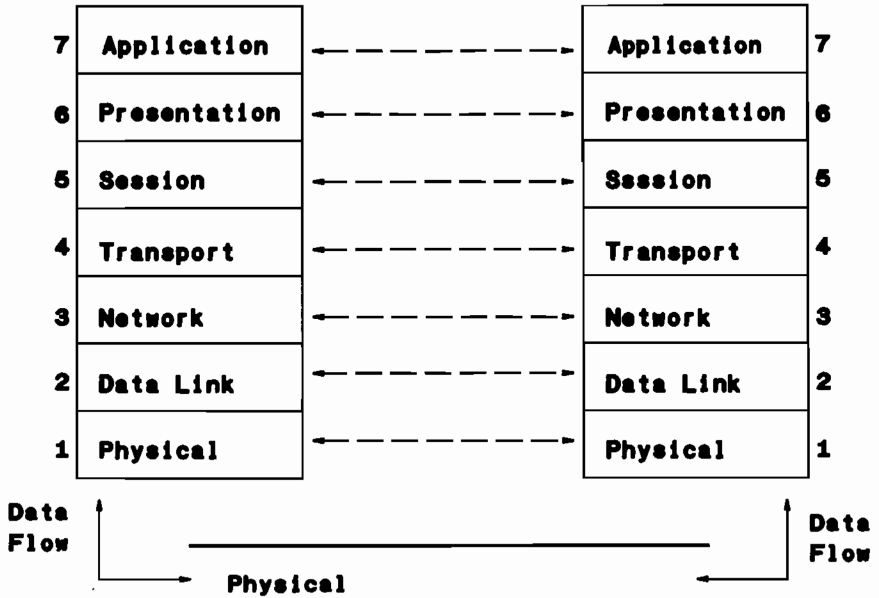


Figure 2

4005-13

OSI PROTOCOL LAYERS
AND
"LAYER SUBSTITUTION"

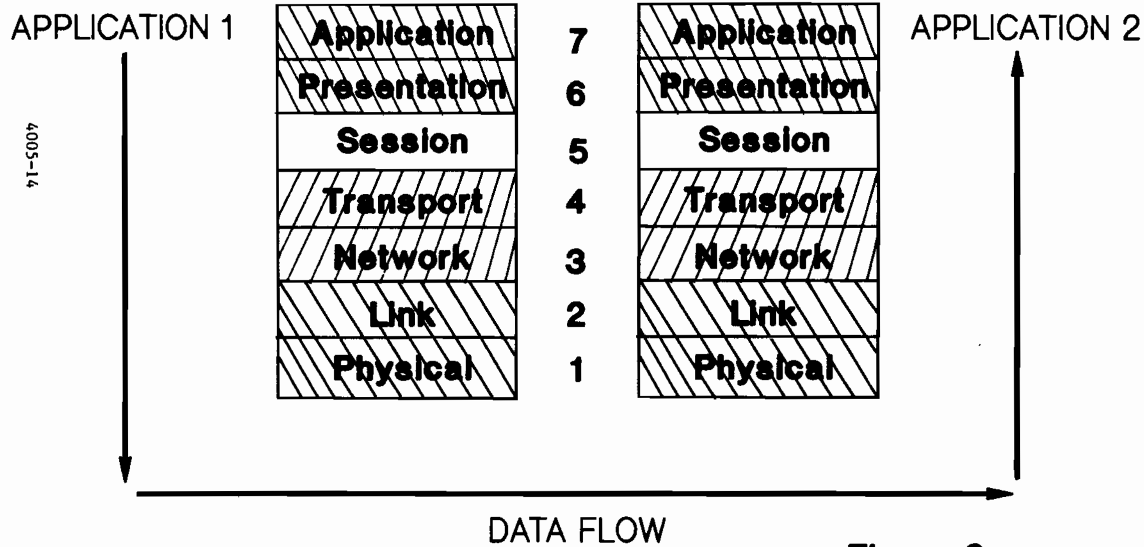


Figure 3

OSI Protocol Stack Comparisons

HP Officeshare		OSI	Novell Netware
AdvanceLink	7	Application	User Software
VT	6	Presentation	User Software
NETIPC	5	Session (API)	Novell NETBIOS
TCP	4	Transport	SPX
IP	3	Network	IPX
802.3	2	Data Link	802.5
802.3	1	Physical Link	802.5

Computer Museum

Figure 4

HP Novell Gateway

Protocol Stack Relationships

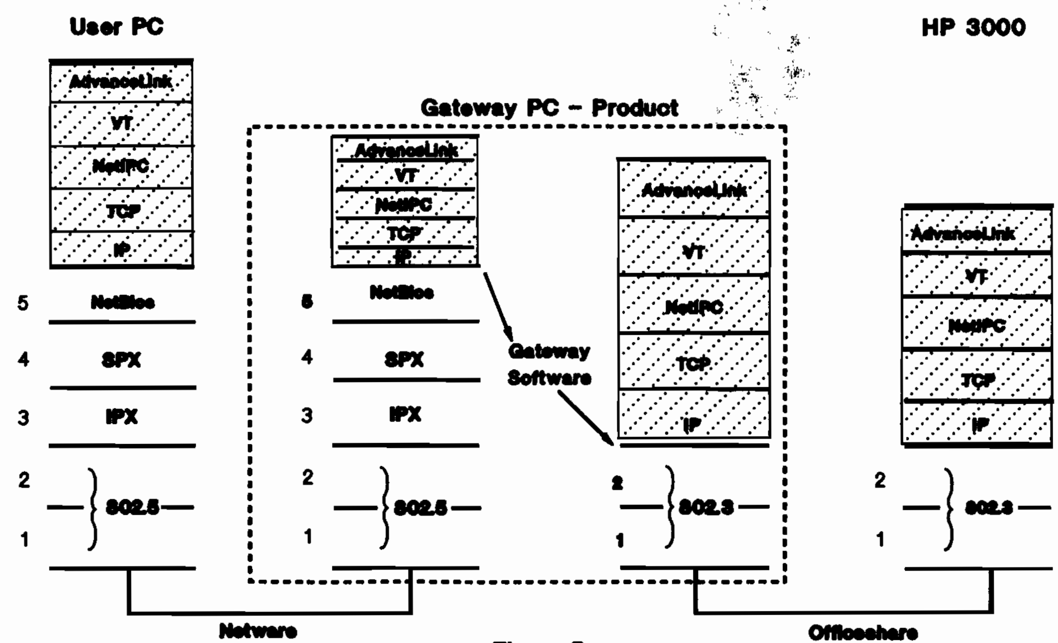


Figure 5

TITLE: A Cooperative Processing Strategy for the 90's

AUTHOR: Doug Walker

Walker, Richer & Quinn

2825 Eastlake Ave.

Seattle, WA 98102

206-324-0407

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 4006

Background Operation of Electronic Mail within the Hewlett-Packard NewWave Environment.

by

David R Morris

Hewlett-Packard Company
Pinewood Information Systems Division
Nine Mile Ride
Wokingham
Berkshire
England

Introduction

This paper is intended to explain the difficulties, and rewards, of implementing an application that performs extensive background operations within the Hewlett-Packard(HP) NewWave environment. It is based on my experiences of implementing the transport system of HP NewWave Mail, but the design approach and technical principles portrayed in this paper are equally applicable to any background application.

The paper is targeted at application developers interested in developing an application for background operation, or technical users who are interested in the working mechanisms of a background application.

The need for background activity

Many user applications available today consist of an interactive user interface and a non interactive 'engine'. The 'engine' will be responsible for performing certain operations that do not provide any user interaction, but prevent the user from performing any further interactive operations. In the case of an electronic mail application, the interactive user interface consists of the ability to send and receive messages, and the 'engine' is the transfer system that exchanges the mail messages with the external electronic mail server. Increasingly, users are expecting the 'engine' component of an application to perform non-intrusively, allowing the interactive user interface to remain responsive.

The ability to maintain a responsive user interface is not a major problem on computers with powerful operating systems that allow many applications to execute (multi-tasking), and prevent any applications from consuming too much resource before another application is allowed to execute (preemptive). Unfortunately this is not the case with personal computers running MS-DOS, which is a single tasking operating system and does not provide the ability for multiple applications to execute (with the limited exception of TSRs). An extension to MS-DOS is MS-Windows. This provides a graphical user interface, mouse as well as keyboard input, and the ability for multiple applications to execute concurrently. These multi-tasking capabilities are non preemptive as they do not prevent any application from consuming large amounts of resources.

The HP NewWave Environment, based as it is on Microsoft Windows, presents the application developer with the ability to provide the user with an interactive user interface, and an active 'engine'. To achieve this, the application has to be carefully designed and implemented.

Background processing facilities

To develop a NewWave application, the developer needs a MS-Windows software development kit (SDK), and a HP NewWave developers kit. The MS-Windows SDK provides about 450 windows function calls, and the NewWave developers kit adds about the same number of object management facility(OMF), application programmers interface(API) and utility function calls. The HP NewWave developers kit does not provide any additional features to MS-Windows that help an application to be non-intrusive to the user, infact a large number of the OMF and API calls result in a very intrusive operation.

As MS-Windows is a non preemptive multi-tasking system, the executing application has to release the processor to enable other applications to perform some processing. This release of the processor is known in MS-Windows terms as a yield. Whenever an application yields, the MS-Windows kernel will allocate the processor to another application. At some time in the future, the application that yielded will again be given the processor.

A yield can be caused either explicitly or implicitly by an application. An explicit yield occurs when one of the following MS-Windows function calls is made :

PeekMessage
WaitMessage
GetMessage

An implicit yield occurs when a MS-Windows function call is made that requires MS-Windows to display another window, such as :

DialogBox
MessageBox

The whole process of yielding is important to multi-tasking in MS-Windows, but is key to the success of a background application. It must be tightly controlled to remain non-intrusive to the user, and prevent problems such as re-entrancy. Re-entrancy occurs when an application receives a message from windows, when it is already processing a message. This can lead to sections of code being executed multiple times, or out of sequence and, if not controlled, can result in system failures.

Needs of an electronic mail system

Having established the capabilities of the platform we are designing the application for, it is necessary to establish the needs of the application, in this case an electronic mail application, to make the possibility a reality.

The basic required features of an electronic mail application can generally be summarized as the following :

- To receive mail from another electronic mail user.
- To send mail to another electronic mail user.

In additional to the basic needs, there is normally a strong desire for additional features such as :

- Checking the distribution list of the mail message to ensure that the intended recipients are correctly addressed.
- Setting a priority on messages that are being sent.
- Informing the user when new mail has arrived.
- Tracking messages that have been sent to determine whether they have been delivered and read by the recipients.

The implementation of an electronic mail application is to a large extent dependent on the target platform. It would be possible to implement all of the required features in a single application that has sub-areas for each distinct action, such as an in tray, out tray and transport system. It would be possible to implement this type of system in the HP NewWave environment, simply by making the application an object, but to make an electronic mail system truly NewWave like, I believe it is necessary to make each sub-area a distinct object.

A possible design for a HP NewWave electronic mail application is as follows :

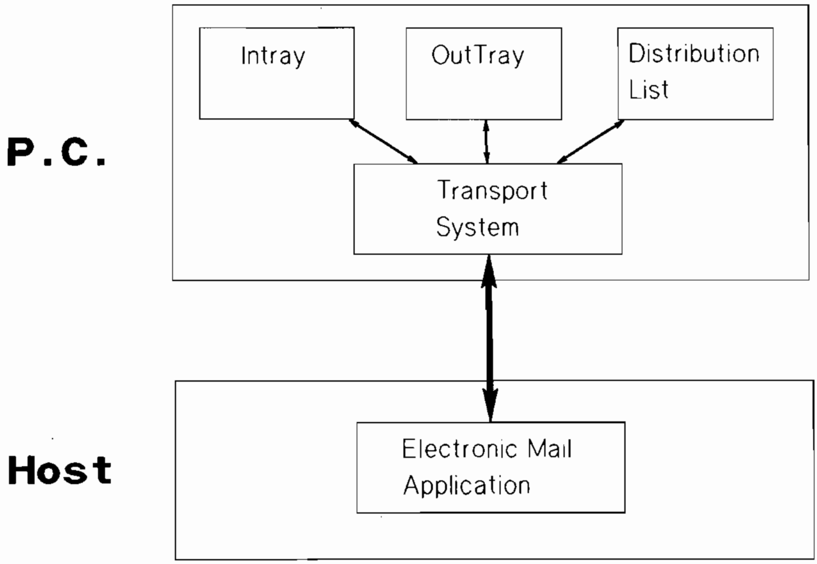


figure 1

Each component of the electronic mail application on the P.C. is an Object. The Intray and Out-tray are global objects, the distribution list is a data object, and the transport is an invisible global object. It is invisible because it provides no user interface, and is a true 'engine'.

Needs of an electronic mail transport system

The needs of an electronic mail transport system can be defined as performing the needs of the user interface, such as sending and receiving mail. For an electronic mail system such as the one shown in figure one, there are some additional requirements :

Inter-object communication
Because each component is an object, there is the need for extensive inter object communication. This has to be handled very carefully to ensure that the transport object does not become re-entered.

Request prioritization
As the transport can receive a variety of requests from a number of objects, it must have the capability to prioritize these requests. The perception of the interactive user interface can be greatly effected by this feature, for example, if a user requests a distribution list check, then this is a high priority task and should be performed immediately, suspending or preempting what ever operation the transport is currently performing.

Transport design

To facilitate all the requirements, it is necessary for the transport ('engine') object to be very well structured. The following diagram shows such a structure :

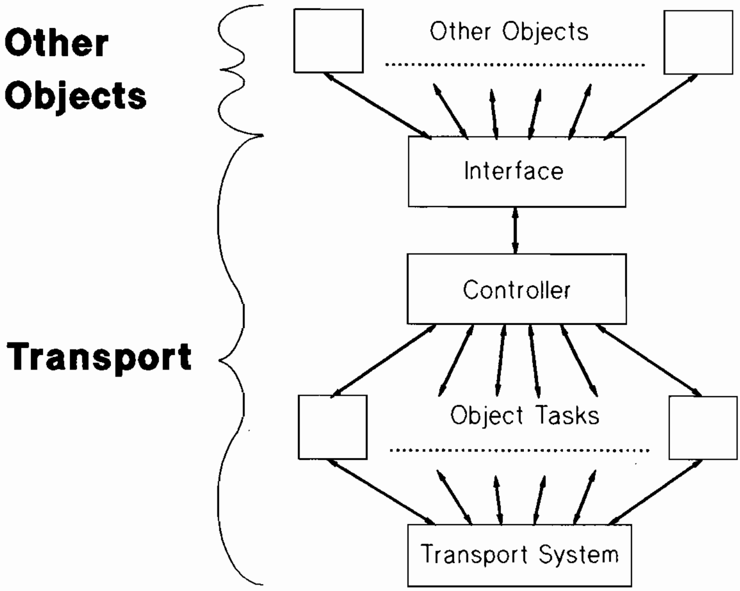


figure 2

The Transport object effectively consists of the interface, controller and several tasks. The transport system is the network connection being used by the personal computer, and can either be a LAN connection or a serial connection.

Interface
This is the highest level of the design and is responsible for all the inter-object communications. In the HP NewWave environment, objects communicate via OMF_SendMessage. This command is generally prefixed by an OMF_GetOMFObject which activates the object if necessary, and suffixed by an OMF_FreeOMFObject which possibly terminates the object. An OMF_SendMessage causes the object receiving the message to get the processor, and the return from the OMF_SendMessage returns the processor back to the sending object. This is synchronous message processing.
This interface level will be able to service some of the requests that are received, but others it will have to pass to the controller.

Controller
This is the key element of the background application as it is responsible for controlling all the lower level operations (tasks), and ensuring the interface level requests are actioned accordingly. This level is effectively an application scheduler that allows low priority tasks, such as sending a message, to be preempted with a higher priority task, such as a distribution list check.

Tasks
An electronic mail application consists of many tasks, each one performing a single operation on the host, such as signing on, signing off, sending mail or receiving mail. The tasks have the freedom to decide whether, or when, they can be preempted by the controller, allowing them to reach convenient break points. They also have the flexibility to suspend themselves and request other tasks to run.

Background implementation

The key to having a successful background operation is to have those operations which are not directly initiated by a user action, to yield very often. It is also very important to only perform very small steps between yields. For the system shown in figure 2, the Interface level can be intrusive, as it is only affected by user actions, but the controller and the tasks need to be non-intrusive and strictly follow the small step yield concept. This concept is achieved by the following :

Small Steps
A MS-Windows programmer will be very familiar with the concept of very large switch statements. These are normally implemented to determine the identity and course of action for a windows message just received. The switch default is to call DefWindowProc, a windows function that returns the default for the message that windows expects. This concept can be taken a stage further to achieve efficient background operations by breaking a simple function into a series of steps. Each step is a small action within a switch statement, and an explicit yield is made between each step of the switch. The switch can be implemented as a function being called with an offset, and the function returns with a new value offset which determines the offset it is to be called with next time around. It is possible to nest these functions so a switch offset within one function calls another function with a switch offset that performs another small step and returns with its switch offset. Care must be taken with the explicit yield to ensure that the application is not re-entered. For the reason, and to ensure efficient preemption, the explicit yield is not performed at the low level.

Yielding between OMF Calls
A potential major impact of any background application is when a call has to be made to a library. The application writer typically has no control over the length of time a library call takes. Fortunately, the small step yield approach can be implemented between successive library calls, reducing the user impact to a minimum. This is particularly important for OMF calls that generally are rather expensive. One such operation is writing properties on an object. Property writes are expensive because they are guaranteed to be written to disk, not just to a cache. The following section of code shows a typical implementation of the small step yield principle :

```
function WritePropertyies(noffset)
int *noffset
{
      switch (*noffset) {
      case 1    :  OMF_GetOMFProperties()
                   *noffset = 2;
                   break;
      case 2    :  OMF_WriteProperty(PROP 1)
                   *noffset = 3;
                   break;
      case 3    :  OMF_WriteProperty(PROP 2)
                   *noffset = 4;
                   break;
      case 4    :  OMF_FreeProperties()
                   *noffset = switch_completed_successfully;
                   break;
      default :
                *noffset = switch_completed_in_error;
      }     /* of case */
}
```

NetIPC NPMT
Another library that is frequently called from an electronic mail application is the network library. There are many types of networks available, but this paper concentrates on NetIPC, part of HPs' AdvanceNet network solution. Typically, calls to network libraries are the same as calls to other libraries, except that they are dependent on a network response. This may result in a very much slower response if the network is unable to respond immediately. NetIPC provides a mode of operation intended for MS-Windows applications. It is called Non Preemptive Multi Tasking (NPMT) mode and provides the application with two advantages :

Non Blocked Network calls

When a NetIPC function is called (a network request is made), the network library enters a critical region where no yield must be made. When NPMT mode is enabled, the request will return immediately, returning the library from the critical region and allowing a windows yield. This return from the library indicates the amount of data, if any, retrieved from the network.

Data Fragmentation

NPMT mode causes NetIPC to fragment inbound and outbound data into one link layer packet size. A link layer packet is generally around 1400 bytes. This can result in not all the data being sent or received by a single call, so the application has to verify that the expected amount of data has been sent or received. In the case of receiving a buffer of unknown size, it is necessary to parse the buffer to determine whether or not it is complete. Most electronic mail applications have their own client-server protocols and NPMT mode can be used to provide effective data transfer underneath these protocols.

The YIELD

Having concentrated on implementing the small steps, it is very important to have an efficient yield mechanism. It is possible to yield by returning to the GetMessage loop, and waiting for a timer message to be received from Windows. This would then prompt the application to perform another small step before returning to the GetMessage loop and yielding again.

Additional Features

The small step yield design and implementation method provides additional benefits. If, for some reason, it is necessary for the background operation to be performed at a higher priority, then it is possible to remove the yield (not call PeekMessage). This prevents MS-Windows from re-scheduling any other applications and allows the background application to complete in the minimum possible time. This is very useful for implementing a High priority configuration option.

The 3 layer design also provides a great deal of flexibility, providing the controller and controller/task interface are well designed, for adding new functionality in the form of new tasks.

Potential Pitfalls

The small step yield principle does also introduce several possible pitfalls :

Memory moves
One important point to remember is that when an application yields, the MS-Windows memory manager may move memory segments to satisfy the needs of another application. If this is between small steps, then it is very important that the small step is not relying on pointers that may have become invalid. The general approach in assuring against this should be the standard windows programming approach of performing a local/global alloc for the memory required, and locking this handle when the data is required. It is important to unlock the memory before a yield to allow the memory manager maximum flexibility for servicing the needs of other applications.

Error detection
A large proportion of an application should be error checking and recovery code. In a standard application this can become quite complex, but in a background application there are a number of new problems. The application designer has to make several decisions as to whether to attempt the recovery in the background, or whether to make the recovery a foreground operation. If the latter is decided then the decision has to be made as to whether the user should be informed of what is happening, and if so, how should the user be informed.

Status reporting
Not only is the designer faced with the problem of how to report errors to the user, but there is also the problem of how to report results to the user.

Conclusion

I believe the small step yield concept, and the ability to preempt tasks within an application result in a very powerful and flexible system that exploits the capabilities of MS-Windows multi-tasking, and the HP NewWave environment, to the full and present the user with a consistently interactive user interface. I also believe the design principles are applicable to any would be 'engine' application.

Buddy, Can You Spare Some Time?:
Using PCs as HP3000 Co-processors

George T. Blessing
City of Pasadena
100 North Garfield Avenue, B-34
Pasadena, CA 91109-7215
(818)405-4220

Abstract

During the last five years, the power of the PC has increased dramatically. Offices now have a vast, almost untapped resource idling the night away.

How could these PCs be used as an extension of the HP3000? We often concern ourselves with ways in which

the minicomputer can serve the personal computer such as backup and print spooling. What about the other side of the coin? Why couldn't the PC serve the mini instead? What can your PC do for your HP3000?

This paper will search for the answer to these and other questions by exploring the uses of PCs as a cooperative processing resource for the HP3000. Of particular concern will be the use of the PC for distributed printing/plotting and as a processor for segmentable problem solving techniques. The paper will document several functions that were tried and evaluate how effective they were in tapping the PC's night time CPU cycles.

The PC explosion has turned the workplace upside down in a matter of years. In our offices alone the number of PCs has increased in the last four years from under 200 to over 700. The sad truth, however, is that in most organizations there are PCs that are poorly utilized. Many executives have one on their desk that they do not even know how to use. A few years ago a PC on a manager's desk might have been a status symbol - now the absence of one could be a social embarrassment. I have found that in our organization, we now have about 3 PCs for every 5 full-time employees. If the current rate of growth continues, we will reach parity within another two years.

I began to hear discussions a few years ago on the topic of using PCs as an additional resource for overnight processing, and the idea intrigued me. At that time, though, there were some serious limitations that stopped me from pursuing the idea. Most of the PCs in our organization then were not IBM compatible and had relatively slow processors in comparison to what was on the horizon. About 80% of the machines only had floppy discs, and the average size of the hard discs was about 10 megabytes. In addition, they were connected to low quality, low speed, dot matrix printers. Many of them were not connected to the HP3000 computers by hardwire, but were connected by modems at only 1200 baud.

Now the average PC here is an 80286 machine with a laser printer (either dedicated or spooled) and 20 megabytes of hard disc. There are many "power-user" machines (at least by last year's standards) with 80386 processors and several hundred megabytes of disc space. Still there is the problem of connectivity - most of the machines are connected through a data PABX at rates up to 19.2Kbps. Even with the newer and faster datacomm programs, a file of significant size can take a long time to transfer between the two devices.

With the power and capacity of PCs reaching the bottom end of the HP3000 line, though, I felt it was again time to consider the possibility of using PCs as peripheral processing devices. I hoped that eventually, the datacomm problem would go away. After all, even as I thought about the situation, the face of the PC and communications marketplaces was changing rapidly. I resolved to investigate four areas that I felt might be worthwhile: printing, plotting, analysis, and problem solving.

Printing

The first area that I examined was the potential for using the PC as a distributed printing device. This was a success, at least from the standpoint that it worked with relatively little effort. It was, however, a failure in that it was not spectacularly fast. I merely started up one of the popular HP terminal emulation packages and logged the output to the attached printer. Overnight processes were modified to write their output to the terminal and the output redirected to the printer.

Observing the process in action, however, showed that the PC was actually hindering the HP3000. Because of the speed of the printer, and the overhead of the datacomm software, a large amount of time was spent handshaking between the PC and the HP3000. (Back to the drawing board.)

The next stab in the printing environment was to log the output to the hard disc instead of the printer, and then print the output afterward. This was a significantly more profitable venture. The speed was such that the HP3000 was seldom waiting for the PC to catch up, and the PC could do the actual printing on its own time. OK, now what?

I found that some reports that required multiple copies could be printed well on a laser printer attached to the PC. By embedding an escape sequence into the transmitted data, the intelligence of the laser printer could take care of making multiple copies of each page without retransmitting the data. This worked good especially for short reports of 1-2 pages, but since the duplicated output is not collated, large reports done in this manner were not fun to separate.

By jumping through a few extra hoops, I could also send name and address information to the PC and have it print form letters all night. There was no need to have someone's PC tied up while they were at work. All that needed to be done was to work out the details and incorporate the procedure into a batch file on the PC. I also found that I also could use preprinted forms in the laser printers, and that the cost of having them printed was significantly less than the cost of continuous forms. The saddest thing I noticed about the output from these printers was that it exceeded by far the quality of our huge (and expensive) system page printer.

Plotting

The next area I considered was plotting. We have a variety of plotting devices here, from 2 to 8 pens and paper sizes A to E. Most do not have auto feeders and this makes them pretty impractical for unattended use. I began this segment by using a similar method to the original printing solution. I logged the output to a plotter and had the HP3000 send HP-GL commands to the PC. As in the printing discussion, it had some obvious drawbacks. (If I thought that the HP3000 had to wait for a printer, imagine how long it waited for a plotter.) I followed my own example and quickly switched to a scheme of logging the output to the hard disc and then plotting it later.

This still really wasn't a good use of a resource. The HP3000 continued to do all of the grunt work, figuring out the HP-GL commands to send. The next approach was to send only the data used to create the plot to the PC, and then use PC software to generate the plot. This was much more beneficial and the solution which I ended with. I was able to eliminate much of the drudgery of some periodic graphics work by performing the work overnight.

I also realized the potential for using printers for graphics output. There are now many software packages and printer cartridges available that make laser and ink jet printers look like HP plotters. We have been able to plot to printers both in black and white, and color. Again, we reaped some of the benefits of the benefits of intelligent printers and plotters by having multiple copies plotted automatically for distribution in the morning.

Analysis

Because my main area of concern at the City is the management of the HP3000 environment, my thoughts about the PCs drifted towards system management applications. One of the first things that occurred to me was that I might be able to reduce the amount of textual information I review periodically. For example, every morning after most of the overnight jobs have completed, we use the HP utility FREE5 to see how fragmented the space is on the disc drives. It wasn't much to look at when we only had a few drives, but now we have 12 to examine.

I had already decided that on a day-to-day basis all I cared about was the amount of free space on each drive and the total for the system. So I had modified the job stream that takes the measurement to eliminate the fragmentation detail and send just the freespace totals to me by way of the electronic mail system. But once a week, I still reviewed the entire output to evaluate the condition of the storage media.

This takes a while and is a little uncertain because the figures may not be all that meaningful at times. After much thought, I arrived at several conclusions about the usefulness of that information. (No doubt many of you will disagree with me.)

1. Any fragment smaller than the average size of all files built during the last 60 days is virtually worthless.

2. All fragments larger than the largest file built during the last 60 days are totally usable.

3. Fragments which are somewhere in between the worthless and totally usable, are usable to a percentage.

The third is probably the least valid, but it has to fit in there somewhere. This worked well for me, but was time consuming. To improve upon my job stream I added an additional step which passes over the directory and extracts those two numbers that are critical to my theories: the average size and the maximum size of files created during the last 60 days. Then I decided to incorporate the PC into the process.

In the middle of the night, at the end of the job stream, I drop all of the information into an MPE file. A little while later, the file is transferred to the PC, and control is given over to a batch file on the PC. A popular spreadsheet program runs and loads a macro-laden worksheet. The information from the MPE file is imported, parsed, and evaluated, and a few moments later a graphical evaluation of the condition of the twelve drives drops into the output tray of my laser printer for my perusal in the morning.

Another area which I am planning to work on, but have not yet attempted, is database analysis. I feel that a PC could be used to analyze the loading of Image master sets and predict what capacities would produce the best placement. This could be done by transferring only the key values from a master set to the PC and then performing the evaluation there. A range of sizes could be appraised and a report of the statistics of each size produced. It could require massive amounts of processor time, but who cares if nothing else is going on anyway.

A PC could also be used to predict the growth and loading of Image databases. The databases on the HP3000 would be examined with a any on of a variety of utilities and the capacity and loading information sent to the PC. On the PC the information could be loaded into a relational database. Periodically, an analysis could be performed and growth projections made to estimate when a database would need expansion or reorganization.

Problem Solving

Because the PCs have a lot of time on their hands, they are available to solve problems which we would not normally consider to be feasible. For example, using trial and error to solve problems with multiple variables for the best possible solution within given parameters. As I have mentioned before, if the PCs are not doing anything else during the night, there is no waste in picking their brains.

There are many PC based statistical packages available today at very reasonable prices. If you've priced software for your series 70 lately, you know that it is often not practical to purchase high-powered software for limited use. Certainly a well-equipped PC could be used to perform statistical analyses on data sent down from the HP3000. If only a limited amount of this type of work was being done, performing it on a PC would save many thousands of dollars. In some cases, you could buy the PC with the savings and still come out ahead.

It is possible that very large problems could be solved by using more than one PC. By employing "divide and conquer" methods, problems could be split into manageable pieces and divided among several machines. As each machine reached the solution for its part, it would return the solution to the HP3000 to be recombined (and redistributed, if necessary) for the final solution. I feel fairly certain that high-powered PCs are substantially more proficient at tasks like multiplying or inverting matrices than an HP3000.

Job Assignments

To manage the distribution of nightly work, there needs to be some consistency within the process. To facilitate this, each PC has an identical batch file to start overnight operations making at least that portion of the process easy to maintain. Each evening the HP3000 prepares a DOS batch file for each PC it has available, even if it is a dummy file which specifies no actions. Each PC executes its overnight operations batch file which begins by loading datacomm software, making a physical connection to the HP3000, and logging on. The "assignment" file is then downloaded to the PC, and the PC logs off, and terminates the datacomm program. The batch file then transfers execution to the assignment file and continues. In this way the scheduling for what is to be done by each PC overnight can be maintained in one place.

I have divided the PCs into three loosely defined classes for the purpose of defining appropriate tasks:

<u>Small PCs</u>
These are typically 80286 machines running at 12MHz or slower with 20 megabytes of hard disc. Small PCs could be used for printing of short reports, forms, and letters.

<u>Medium PCs</u>
I consider medium PCs to be at least an 80286 processor at 16Mhz with hard disc space greater than 35 megabytes. Medium PCs can be used for light to medium complexity graphics processing, analysis, and forecasting.

<u>Power User PCs</u>
These are PCs with 80386 or faster processors, math coprocessors, and large amounts of available disc space and memory. Power user PCs should be used for complex 2-D graphics processing, 3D visualizations of data, and large analytical problems.

The Future

In the future I hope to see significant improvements in how PCs can be used in these ways. One of the enhancements on our horizon here is the advent of a 10Mbps local area network. This should make some of the transfer time problems go away and increase the variety of solutions for high-volume data analysis. I expect to see this as a reality here before INTEREX 90. I hope to even see multiple-PC solutions to problems being entirely controlled by one of the PCs on the network.

In addition to improved datacomm speed, I believe that we are about to see an expansion in the area of artificial intelligence. Not only could a PC be used to analyze massive amounts of performance data, but it could also have access to a knowledge base filled with past observations and rules.

Someday, I expect to be able to come into my office in the morning, and find a letter in my electronic mail from the system providing me with only the information about the system I need to know, and recommendations on how to improve it. God willing, that day is not too far off.

Cooperative Processing Today with PC's, Macintosh's, and HP Minicomputers

Ron Arons
Sybase, Inc.
6475 Christie Avenue
Emeryville, CA 94608
(415) 596-3500

Background

In most corporations today, MIS organizations are trying to determine how they can derive more productivity from their computer systems. Unfortunately, in many cases, these organizations have a myriad of diverse computer hardware systems, each with its own set of application software and related data. These so-called "islands of information" are a source of many MIS problems including:

- replicated sets of data, not all of which are consistent with each other
- end-users having to rekey information from one system into another system
- each computer hardware/software system not necessarily being used for what it does most effectively
- needing additional MIS personnel to administer systems that are not being used to their full potential.

Fortunately technology exists today to not only eliminate these problems, but also to allow for some exciting new applications to be developed. This paper will examine the theory behind the new technology as well as give three real life examples of how it is being put into use.

Theory of Cooperative Processing and Client/Server Computing

Cooperative processing, simply stated, is the ability of multiple application programs to work with each other in real time. The term cooperative processing usually refers to tying together applications that reside on diverse hardware systems, although cooperative processing can take place between two or more applications on the same computer system. The concept goes beyond terminal emulation and file transfer to a notion of multiple computer systems working together as one logical system. The computer hardware systems may be from the same or different hardware vendors; similarly the software running on these systems may be from the same or different software vendors. What is important to note is that ALL of the hardware systems and associated software applications can work together as an integrated whole.

"Client/server Computing", takes the notion of cooperative processing one step further. The client/server model of computing asserts that "clients", usually PC's, Apple Macintosh's, UNIX workstations or character terminals, are especially good at providing the graphical user interface portion of an application to an end-user (human being). A "server" system, usually a host system on the order of a HP 9000 or HP 3000/XL minicomputer, is optimized for "serving" multiple clients and for handling the non-end-user oriented computing activities such as backup, security, data storage, and disk I/O. By combining the strengths of each, i.e. the ease-of-use of a PC or bitmapped workstation and the power of a minicomputer "server", corporations can leverage the investment made in all of these different computing systems. Additionally, the same technology allows new exciting applications to be built that would not have been possible with just one of the individual hardware/software components by itself.

A Closer Look at the Cooperative Processing (Client/Server) Technology Available Today

The technology required to tie diverse computer hardware systems together, fortunately, is available today. On the client side, there are client application programming interfaces (API's) that are being used to facilitate the use of many popular PC productivity tools, e.g. dBASE, 1-2-3, HyperCard, with software running on host systems, i.e. relational databases and other server software applications. These same API's can interface custom applications, e.g. written in 'C', to work with server software running on the host systems.

On the host system side, the technology exists for relational database systems and other non-database server software to communicate with each other via "server" application programming interfaces. These host based API's allow cooperative processing with a variety of server software systems, such as real-time data feeds, mail systems, and 'compute servers'.

Client and server hardware/software systems are tied together via a common network protocol such as TCP/IP, LAN Manager or DECnet. All systems use the same network protocol to communicate with each other over a local area network (LAN) or wide area network (WAN).

The following case studies show how corporations are implementing this technology today.

Case Study #1: U S West Direct

U S WEST Direct is a division of U S WEST, one of the Regional Bell Operating Companies (RBOC's) spun off from AT&T in the divestiture of MaBell's local operating entities. U S WEST Direct's mission is to provide Yellow Pages advertising service for 14 states in the Midwest and Northwest via 300+ telephone directories that the company produces annually.

Historically, U S WEST Direct has used a Bull DPS 90 mainframe containing nearly 9 GB of data. The company is currently implementing a new strategy of using the Macintosh as the "front-end" workstation of choice for end-users and HP 9000/300 minicomputers as "back-end" servers to 4-5 Macintosh systems. This configuration will be replicated in approximately 80 "systems" at about 40 sites in the U S WEST Direct territory. The goal of using the Macintosh/HP 9000 combination is to decentralize operations making each of the local teams more efficient in their operations and less reliant on the central Bull mainframe system. All sites will be connected together via a X.25 wide area network (WAN).

This hardware/software configuration will be used for three related applications:

1) a sales order entry application for taking orders for advertising in one or more Yellow Pages directories.

2) contract and advertisement management. This application will keep track of all the details of a particular client's advertising contract terms.

3) a decision support tool for reporting to U S WEST Direct management, in real-time, how well the operation is functioning. This latter application has already been developed with 4th Dimension, one of the leading Macintosh database/application development systems.

Each regional "system", in summary, will consist of several "front-end" applications on the Macintosh that will be cooperatively processing with a database engine running on the HP 9000 minicomputer. The Macintosh is being used because of its end-user friendliness, while the HP 9000 systems are being used for their power, reliability, and ability to support large sets of data (for both on-line and backup purposes).

A major future enhancement planned for the future is the use of the HP minicomputer for central storage of advertisement text and images created on the Macintosh. This will be an improvement over the current method of retaining ads in Macintosh files, backing up these files to cartridge tapes, and sending the tapes to a central publishing center.

Case Study #2: Defense Advanced Research Projects Agency (DARPA)

The Defense Advanced Research Projects Agency is currently implementing a new system that will also include both Macintosh's as end-user workstations and a HP 9000/855 system as a central server. The agency is planning to build custom applications to handle the entire agency's financial activities.

Specifically, the Macintosh/HP 9000 system will be used for several end-user applications:

- long term planning and budgeting. This includes programming funds to support DARPA research initiatives.

- budget execution on a daily basis. This includes tracking funds actually "committed" or "obligated".

- tracking actual agency "expenditures"

The Macintosh is the standard workstation used by DARPA's 170 employees and outside support contractors. The easy-to-use

Macintosh interface is important for providing access to "infrequent" users of the system. (DARPA program managers typically travel a fair amount, so they tend to forget how to generate complex data queries.) Up to 60 concurrent users of the Sybase database running on the HP 9000 are expected. Approximately half of those will be daily users of the system, while the remainder will be infrequent users. (Average access is in the range of once a week to once a month.) The front-end applications will be built in 4th Dimension or HyperCard, two application development systems available for the Macintosh.

The front-end tools (4th Dimension and HyperCard) were selected for their compatibility with the standard DARPA workstation and because they provide an easy access method for users with a wide range of computer experience, ranging from the complete novice to very advanced computer users. Communication with the HP 9000 via Ethernet and TCP/IP network protocols.

The HP 9000 was selected because it is an "open" UNIX system and will allow a smooth transition from the "closed" MPE operating system currently in use. Migration to a "portable" operating system is a long range agency goal.

DARPA selected Sybase for its back-end database on the HP 9000. Data checking had to be automatically and consistently applied across all database update modes. Other database systems were found to be unacceptable because they either required duplication of data checking code or lacked data checking facilities. DARPA determined that only Sybase was capable of meeting DARPA's strict requirements for data integrity and data security.

Cast Study #3: American Express Life Assurance

AMEX Life Assurance is currently (5/90) planning to tie Macintosh systems to either an HP 9000, VAX or Stratus minicomputer system.

AMEX Life Assurance is in the process of revamping its computer systems for policy processing. The company traditionally has kept many of its insurance policies on an Amdahl mainframe. Since early 1989, the company has stored new accident insurance policies on a minicomputer system. The company is currently using Macintosh's as front-end workstations to a minicomputer system (HP 9000, VAX or Stratus are the key contenders at the time this paper is being written), to work on new policies as well as access older policies still stored in the Amdahl mainframe system.

AMEX Life is using the Macintosh to provide:

- customer service for accident insurance policies (this involves accessing new insurance policies stored on the minicomputer or other policies still located on the Amdahl mainframe)
- word processing
- spreadsheet work
- mail

HyperCard, a software package that comes bundled with every Macintosh, is the development system being used to create the end-user interface of the policy processing program. HyperCard was chosen for several reasons:

- it runs on the Macintosh, the workstation that everyone at AMEX Life uses
- HyperCard is a very good prototyping tool
- it is easy to learn
- the programming language in the product is easy to manipulate
- the program is "quasi-object oriented"

The HP 9000 is being considered as the server system of choice for several reasons:

- HP's willingness to work with the customer
- the power of the high-end HP 9000 computer line (50 MIPS with the forthcoming HP 9000 Model 870)
- HP's product line is considered to be "at the start of the power curve" and much more at the leading edge of technology than other minicomputer vendors' offerings.

Like U S WEST Direct and DARPA, AMEX Life would connect Macintosh workstations to the HP 9000 via TCP/IP network protocols.

Other Possibilities Today

In all three cases, the customers are using Sybase software to perform cooperative processing between Macintosh systems and HP 9000 minicomputers. Application programming interfaces, that are identical to the Macintosh API's discussed in this paper, are also available today on PC's running either DOS or OS/2. These PC-based API's provide many of the popular PC productivity packages with the ability to access data in a Sybase database running on the HP 9000 system.

Future Possibilities

Sybase is also working on a version of its DBMS system that will run on HP 3000/XL systems; this version will be available late this year or early next year. Using the same applications on the Macintosh and PC's, corporations will have the opportunity to have cooperative processing between PC's and Macintosh's and their HP 3000/XL systems.

Another possibility in the future will be the choice of networking options. While Macintosh and HP minicomputers currently utilize TCP/IP facilities to communicate with each other, LAN Manager (and its relatives such as LM/X (LAN Manager for UNIX) will soon provide an alternative network protocol to tie these diverse systems together.

Summary

The technology exists today to coordinate the facilities of one hardware/software system with those of other systems into one integrated whole. The benefits of this technology include:

- new, synergistic applications that take advantage of each hardware/software system's particular strengths

- cost savings/increased productivity: because different hardware/software systems can be used together as one logical whole, there are cost savings to be reaped. At minimum, data does not have to be replicated on multiple systems; all systems can leverage the same data. Productivity of end-users is increased as they are ability to do more with the same amount of hardware/software.

TITLE: Costing PC Integration..Is it worth it?

AUTHOR: Jim Wowchuk

Vanguard Computer Services

93 Moncrieff Dr.

East Ryde, NSW

Australia

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 4035

CREATE A DATA DICTIONARY ON PC

PHIL NGUYEN
LOCKHEED ENGINEERING AND SCIENCES CO
2400 NASA ROAD 1 HOUSTON TX 77058
(713)-333-7177

I WHY A DATA DICTIONARY IS NEEDED ?

MOST SYSTEMS LACK ADEQUATE DOCUMENTATION. USUALLY THE PROCESS IS NEGLECTED BY THE IMPLEMENTER SINCE IT IS NON-CREATIVE, TIME CONSUMING AND IS NOT PERCEIVED HIGH PRIORITY BY THE CUSTOMER. IN MANY ASSIGNMENTS, A PROGRAMMER IS OFTEN GIVEN INPUT DATA, WHAT OUTPUT SUPPOSED TO BE AND IS FREE TO PLUG IN ANY FORMULA TO BE ABLE TO COME UP TO THE DESIRED OUTPUT. SUCH A DEVELOPMENT APPROACH TAKES TIME AND IS ERROR PRONE.

WHEN MAINTAINING A COMPLEX SYSTEM, THE PROBLEMS ARE COMPOUNDED BY TURNOVER OF PERSONNEL. AS EXPERIENCED PEOPLE LEFT, KNOWLEDGE ABOUT SYSTEM IS NOT ASSIMILATED BY THE REST OF DP STAFF.

DATA DICTIONARY IS ALSO NEEDED TO BE ABLE TO EXPLAIN TO USERS AND INSTILL IN THEM THE CONFIDENT THAT ALL OUTPUT DATA IS CONSISTENT AND CORRECT. AS SUCH, IT HELPS IN THE DAY-TO-DAY TASK OF DATA VALIDATION.

II SYSTEM DESCRIPTION

THE PC DATA DICTIONARY WILL

- GENERATES HIGH QUALITY DOCUMENTATION
- MAINTAINS DESIGN CONSISTENCY
- STORES AND CONTROLS DATA AND PROCESS DEFINITIONS
- PRINT DATABASE SCHEMA AND DATA DEFINITIONS
- TRACKS SYSTEM COMPONENTS AND STRUCTURES

THE REPORTS MENU OFFERS 20 DIFFERENT REPORTS INCLUDING REQUIREMENTS, PROCESS FLOW, AND A VARIETY OF CROSS REFERENCES, DEFINITION AND DIAGRAM LISTINGS.
THE MAINTENANCE MENU SUPPORTS RE-INDEXING OF THE DATA BASE FILES, ADD OR CHANGE ENTITY NAME, UPDATE DEFINITION AND RELATIONSHIP.

III DESIGN AND IMPLEMENTATION

APPLICATIONS RUNNING ON HP 3000 USING COGNOS POWERHOUSE PRODUCTS LIKE QUICK, QUIZ, QTP. POWERHOUSE MAINTAINS A DICTIONARY TO CONTROL VARIOUS SYSTEM COMPONENTS.

A. PREPARATION STEPS

1. USE ARCHITECT

RUN ARCHITECT TO DOCUMENT YOUR APPLICATION. THIS SOFTWARE WILL CREATE AN IMAGE DATA DEFINITION DATABASE (DDDB). THE DDDB CONTAINS DEFINITION OF FILES, ELEMENTS AND PROGRAMS AND THEIR ATTRIBUTES. IT KEEPS TRACK OF EVERY ENTITY IN THE SYSTEM AS WELL AS PROVIDING RELATIONSHIP BETWEEN FILES AND ITEMS

ARCHITECT ALSO OFFERS :

- ON-LINE CROSS REFERENCING OF FILES, ELEMENTS, PROGRAMS
- ON-LINE APPLICATION MAINTENANCE
- AUTOMATIC DOCUMENTATION OF COMPLETE APPLICATIONS

AFTER THE DDDB IS BUILT, USE ARCHITECT SCREEN TO UPDATE DATA DEFINITION

IF YOU DO NOT HAVE ARCHITECT THEN A SOFTWARE WITH SIMILAR FUNCTION WILL DO

2. USE QUERY

WHEN YOU ARE SATISFIED WITH THE CONTENT OF THE DDDB, USE QUERY TO LOOK AT THE INDIVIDUAL DATA SET AND DETERMINE WHICH DATA YOU WANT TO DOWNLOAD TO PC

YOU MAY WANT TO DOWNLOAD THE WHOLE DATA SET OR JUST A NUMBER OF FIELDS IN THAT DATA SET

BEFORE DOWNLOADING DATA, WE CAN RUN A COBOL PROGRAM TO FORMAT THE DATA SO THE FILE IS READY TO BE IMPORTED INTO PARADOX TABLE

INSIDE QUERY, YOU CAN USE "SAVE" COMMAND TO KEEP THE DATA RETRIEVED IN A FLAT FILE. THEN IT IS EASY TO WRITE A COBOL OR FORTRAN PROGRAM TO FORMAT THE DATA.

ANY AVAILABLE RELATIONAL DBMS PACKAGE ON PC CAN BE USED TO IMPLEMENT OUR DATA DICTIONARY. THE FORMAT OF RECORDS ON FILE DOWNLOAD TO PC DEPENDS ON THE DBMS PACKAGE USED.

3. DOWNLOAD DATA

FILES CAN BE DOWNLOADED TO PC USING "ADVLINK" OR "KERMIT" SOFTWARE

C. <u>APPLICATION ON PC</u>

1. CREATE PARADOX TABLES AND IMPORT DATA FROM IMAGE DATA DICTIONARY (DDDB).

   THE TABLES FALL INTO 2 GROUPS :

   _ CATEGORY 1 DEALS WITH DATA DICTIONARY DEFINITION OF DATABASES, FILES, ELEMENTS AND DATA FLOWS.

   _ CATEGORY 2 PROVIDES DETAIL DOCUMENTATION ON TASKS PROCESSES, PROGRAMS AS WELL AS AN INDEX ON FLOW CHARTS FOR EASY RETRIEVAL

[1]	AAPPL	HP 3000 SYSTEM APPLICATIONS DESCRIPTION
[2]	ANOTE	TO RECORD DATA FLOW WHEN THERE IS MORE THAN ONE SOURCE
[3]	ABMAST	DATA BASE NAMES, DESCRIPTIONS, ATTRIBUTES
[4]	ABDETL	DATA BASE AND RELATED DATA SETS
[5]	AFDETL	FILE AND RELATED ELEMENTS
[6]	AFRELT	RELATION BETWEEN FILES
[7]	AFMAST	FILE DESCRIPTIONS AND ATTRIBUTES
[8]	AEDETL	SOURCE AND USE OF DATA ON ITEMS LEVEL
[9]	AEMAST	ELEMENT DESCRIPTIONS AND FORMATS
[10]	PCODE	TABLE OF FILE CODES
[11]	PDETL	DETAIL DATA MANIPULATION BY PROGRAM
[12]	PMASTR	DOCUMENT OF FORTRAN PROGRAMS
[13]	PREFN	INPUT OUTPUT FILES USED BY PROGRAM

[14] TSKMST TASK CODE AND DESCRIPTIONS

[15] PROMST PROCESS MASTER

[16] PRODEF PROCESS DEFINITION

[17] PRFLOW TEXTUAL DESCRIPTION OF FLOW

[18] DEXPO PROGRAM PSEUDO CODE

[19] PFCHAR PROCESS AND HAVARD GRAPHICS CHART ID

[20] PPREL PROCESS AND PROGRAM RELATION

2. CREATE APPLICATION MENU AND DECIDE ON WHICH FUNCTIONS TO SUPPORT. HOW MANY REPORTS AND SCREENS ARE TO BE INCLUDED ? FOLLOWING ARE MENU TREES FOR BASIC APPLICATION AND FOR THE EXTENDED SYSTEM

Extended package

Application Menu Tree:
=====================

```
Main
        DEFINITION [ DataEntry ]

        FLOWCHART  [View]

        REPORT

                     PROGRAM DETAIL [Report]
                     EXPLAIN [ Report ]
                     STEP [Report]
                     COMPOSITE [Report]

        OTHER

                     PROC MASTER [View]
                     ENTER DETAIL [Data Entry]
                     STRUCTURE [Report]
                     STATUS [Report]

        Leave [ Leave ]
```

Basic package

Application Menu Tree:

```
Main
          REPORT
                      FILE DESC [ Report ]
                      ELEMENT DETAIL [ Report ]
                      SIMPLE SU [ Script ]
                      COMPLEX SU [ Script ]
                      APPL & BASE [ Script ]
                      EXIT [ Cancel ]
          INQUIRE
                       XREF FILE/ITEM [ View ]
                       ELEMENT/FILES [ View ]
                       DATA FLOW [ View ]
                       FILE AND BASE [ View ]
                       FILE DIRECTORY [ Report ]
          MAINTAIN
                        ADD BASE [ DataEntry ]
                        ADD FILE [ DataEntry ]
                        NEW ITEM [ DataEntry ]
                        FILE REL [ Edit ]
                        DATA FLOW [ Edit ]
          UTILITIES
                         DESCRIPTION [ Script ]
                         HELP [ Help ]
                         UTIL PROG [ Script ]
                         XREF SOURCE [ View ]
                         DB MAINT [ Script ]
          PROGRAMS
                        DOCUMENT [ Report ]
                        PROG DETAIL [ Edit ]
                        PROG REFN [ DataEntry ]
                        DIRECTORY [ Report ]
                        COUNT [ Script ]
                        REPORT DETAIL [ Report ]
          Leave [ Leave ]
```

3. TO CREATE VIEW SCREENS AND SIMPLE REPORTS, WE USE THE PARADOX PERSONAL PROGRAMMER _ A PROGRAM THAT HELPS YOU CREATE APPLICATIONS WITHOUT PROGRAMMING. PARADOX, A RELATIONAL DATABASE OFFERS LIMITLESS POSSIBILITIES FOR ORGANIZING, MAINTAINING AND REPORTING INFORMATION.

ALTHOUGH PREPLANNING WILL MAKE THE ACTUAL CREATION PROCESS QUICKER AND EASIER, YOU DO NOT NEED TO PLAN EVERYTHING IN DETAIL BEFORE USING THE PERSONAL PROGRAMMER. IT IS FORGIVING, ALLOWING YOU TO CHANGE YOUR MIND ABOUT HOW YOUR APPLICATION LOOKS AND WHAT IT DOES AT ANY TIME.

THE FOLLOWING EXAMPLE PRESENTS TWO SCREENS THAT CAN BE DEVELOPED QUICKLY. THE FIRST SCREEN SHOWS ALL FILES AND DATA SETS THAT HAVE THE SAME DATA ITEM. THE USER NEEDS TO TYPE IN THE NAME OF THE ITEM, IN THIS CASE, P1-REQ-NO WHICH STANDS FOR P1 REQUISITION NUMBER. THE SYSTEM WILL RETRIEVE AND DISPLAY ON SCREEN LISTING OF ALL FILE NAMES, FILE TYPES AND RESPECTIVE DATA BASES HAVING P1-REQ-NO AS ONE OF THE FIELDS IN THE RECORD STRUCTURE.

NOW IF AN USER WANTS TO FIND SOURCES OF DATA FOR A FILE OR A DATA SET, THEN HE OR SHE JUST NEEDS TO TYPE IN THE NAME OF THE FILE. THE SCREEN WILL DISPLAY ALL INFORMATION RELATED TO THE ABOVE QUERY.

TO CREATE REPORTS FOR THE DICTIONARY, IT REQUIRES SIMPLE PLANNING :

a) DETERMINE WHAT INFORMATION TO SHOW ON REPORT. FIND OUT ON WHICH TABLES THE DATA RESIDES.

b) USE THE REPORT GENERATOR TO DESIGN CUSTOM REPORT SPECIFICATIONS ARRANGING AND SUMMARIZING THE INFORMATION ANY WAY YOU WANT. YOU PERFORM THE FOLLOWING TASKS

- SELECT FIELDS FROM THE TABLE AND FROM LOOK UP TABLES
- GROUP INFORMATION TOGETHER
- CHANGE THE NAMES AND FORMAT OF COLUMNS
- ADD TITLES

IF YOU FIND AN EXAMPLE THAT RESEMBLES THE REPORT YOU WANT TO PRODUCE, TRY IT OUT ON PARADOX TABLES, THEN ADAPT IT TO YOUR SITUATION.

c) AFTER YOU HAVE PUT ALL THE BUILDING BLOCKS IN ORDER, USE THE PERSONAL PROGRAMMER TO ATTACH THE REPORT ACTION TO A SELECTION ON THE APPLICATION MENU. DEFINE AND AUTOMATICALLY CREATE A QUERY TO EXTRACT REQUIRED INFORMATION. TO GENERATE THE REPORT YOU REQUEST, THE PERSONAL PROGRAMMER PRODUCES WELL STRUCTURED CODE AND WRITES IT INTO ORGANIZED PROCEDURE LIBRARIES.

FOLLOWING ARE SAMPLE OF REPORTS DEVELOPED

ENTITY/RECORD/FILE DETAIL
WITH DATA ELEMENTS

Project: IPMIS Version: 1

Entity Name: File Name:
PROJECT TIME CARD FILE WTCPRO65

Location:
LABDATA

Description:
PROJECT TIME CARD FILE CREATED FROM PROJECT COST LABOR DISTRIBUTION (CLPC) BY PROGRAM LDRREF

ELE ID	DESCRIPTIVE NAME	FIELD NAME	FORMAT	KE
1	COMPANY CODE	WP-ID1	X(2)	
2	WEEK END DATE	WP-IDTE	X(6)	
3	EMPLOYEE NUMBER	WP-IEMP	X(6)	
4	DEPARTMENT CODE	WP-IWDP	X(6)	
5	PWO NUMBER	WP-IWPWO	X(10)	
6	REGULAR HOURS	WP-REG	9(9)V9	
7	OVERTIME HOURS	WP-OVT	9(9)V9	
8	DOUBLE TIME	WP-DBL	9(9)V9	
9	ABSCENCE TIME	WP-OVH	9(9)V9	
10	TOTAL HOURS	WP-IATD	9(9)V9	
11	EMP LAST NAME	WP-LNAM	X(14)	
12	INPUT BATCH NO	WP-IBATCH	X(4)	
13	FLAG	WP-IOT	X	
14	SHIFT CODE	WP-ISHFT	X(2)	
15	LABOR TYPE	WP-LTYPE	X(4)	

PROCESS: LABOR DISTRIBUTION

The weekly labor distribution cost accumulation provides input

1. Phase 1 input consists of the unaudited time cards and personnel data base as received by HQMIS. It creates a report containing direct labor hours and overhead labor hours. The report is ordered by dept and employee number and provides data pertaining to employee time cards.

2. Phase 2 input consists of the audited time card data from HQMIS. These data are used to develop project labor hours and costs on a weekly basis. For each performing department, the labor data base will feed data to the contract labor analysis process.

The labor data from HQMIS do not contain transactions generated by the adjustment voucher, cost transfer (CTV) process. Therefore, the labor data base is accurate once per month when the work-in-process file has been processed to update the PWO's affected.

PROG NAME: LETAPE

PROG TITLE: CREATE WIP BASE FOR FYA

AUTHOR: JOHN H HOLTON

DESCRIPTION :
READ HQMIS WIP, NET COMMITMENT FILE AND OTHERS. SUM WIP RECORDS TO ONE RECORD PER PWO. CREATE LABOR FILE AND WIP BASE FOR FYA, LPWO AND OTHER COST ANALYSIS PROCESS

REPORT: AUDIT

SCREEN: N

FILES USED :

WIP	NETCOM
FYCARRY	FYCOIP
FYRCVNP	WIPJOP
LEBKDTRV	LEMWIP
LEMLBR	TRVPWO
LABPWO	

4. TO GENERATE COMPLEX REPORTS, YOU MUST KNOW PAL OR A SIMILAR 4GL. PAL IS A FULL FEATURED, HIGH LEVEL, STRUCTURED DATABASE PROGRAMMING LANGUAGE THAT LETS YOU WRITE SOPHISTICATED PARADOX SCRIPTS AND APPLICATIONS.

THE REASON IS YOU HAVE TO RETRIEVE DATA FROM MULTIPLE TABLES AND BLEND THEM TOGETHER ONTO A REPORT. I PRESENT TWO SAMPLE REPORTS. THE FIRST ONE, PROCESS DETAIL WITH DATA FLOWS, EXPLAIN A PROCESS AND TRACK THE FLOW OF DATA THROUGH OUT THE SYSTEM.

THE SECOND REPORT, MULTIPLE SOURCES AND USES, PRINTS A COMPLETE TEXTUAL DESCRIPTION OF SOURCE AND USE OF DATA ELEMENTS IN A FILE.

100 DATA FLOW DIAGRAMS ARE GOING TO BE DRAWN USING HARVARD GRAPHICS WHICH LINK DATA REPOSITORIES TO ACTIVITY CENTERS. THESE DIAGRAMS CAN BE INDEXED ON PROCESS ID AND EASILY RETRIEVED TO PROVIDE GRAPHICAL NOTATION FOR SYSTEM ANALYSIS.

IV BENEFITS

DURING THE IMPLEMENTATION PERIOD, I SEE QUITE A FEW BENEFITS, BOTH SHORT TERM AND LONG TERM. THE IMMEDIATE BENEFITS ARE AS FOLLOW :

1. GENERATES DATA DEFINITIONS AND DATABASE SCHEMAS WHEN NEEDED. PROVIDES VALUABLE TOOLS TO PROGRAMMERS FOR SYSTEM MAINTENANCE.

2. HELP USERS TO UNDERSTAND APPLICATION BETTER, THUS IMPROVE PERFORMANCE. USERS ARE NOW INVOLVED IN DATA PROCESSING ACTIVITIES AND WORK AS A TEAM TO INTEGRATE QUALITY FINANCIAL REPORTING PROCESS.

3. CREATES DATA FLOW DIAGRAMS THAT HAVE BEEN USED IN VARIOUS GROUP MEETINGS RELATED TO DATA VALIDATION AND SYSTEM REDESIGN.

4. BUILDS A SET OF SYSTEM DOCUMENTATION TO SUPPORT SOFTWARE AUDIT, SOFTWARE METRICS AND PRODUCTIVITY.

THE LONG TERM BENEFITS CAN BE SEEN AS OUR DATA DICTIONARY STANDS SIDE BY SIDE WITH THE ON GOING OPERATIONS AND MAINTENANCE. BOTH OF THEM ARE ACTIVE PLAYERS OR POST CASE TOOLS IN THE SOFTWARE LIFE CYCLE.

BY POST CASE, WE TALK ABOUT THE REVERSE ENGINEERING PROCESS THAT CONVERT SPAGHETTI CODE TO STRUCTURED CODE AND STORING IT IN THE CASE REPOSITORY, WHERE IT IS AVAILABLE FOR MAINTENANCE, EDITING OR REUSE.

THE DATA DICTIONARY APPROACH, WHEN FINISH, OPENS UP NEW CONCEPTS IN SOFTWARE REUSE AND SOFTWARE MEASUREMENT.

PROCESS DETAIL WITH DATA FLOWS

Project: IPMIS Version: 1

Proc ID:	Title:	Program:
LAB	SELECT VALID LABOR RECORDS	LABBY

Access Freq:

Description:

READ HQMIS LABOR TRANSACTION. IF THE CONTRACT PERIOD IS IN THE LIST TO SKIP. THE TRANSACTION IS APPENDED TO A FILE. CREATE PC75 CONTAINING VALID RECORDS FOR CURRENT CONTRACT.

Item: PIDOLLAR File: PROJTAPE

Type	Dir	Other ID	Field Name	Of
FLOW	TO	7	TOTAL COST	LABOR MASTER FILE
FLOW	TO	7	TOTAL COST	PROJECT COST LABOR DIST
FLOW	TO	7	LABOR COST	JOP UPDATE FILE
FLOW	TO	8	ACTUAL COST	ACTUAL HOURS AND COST BY JO
FLOW	TO	6	ACTUAL COST	JO PLAN AND ACTUAL DETAIL
FLOW	TO	14	PROGRAM	SUM UP ME,RATE.COST FOR OMGP

Press [F2] when finished viewing the table
Total records: 6

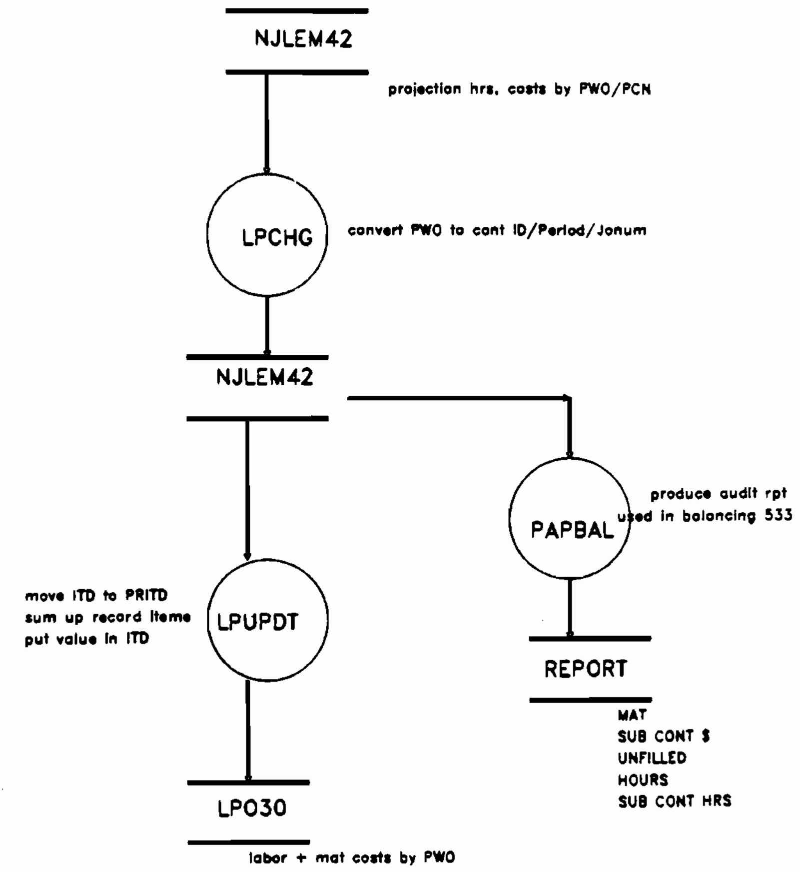
IPMIS DICTIONARY
XREF ELEMENT / DATA FILES

File Name	Organization	Base
APRSBAS	IMAGE DETAIL	APRSDB
APRSDTY	IMAGE DETAIL	APRSDB
APRSOFFR	IMAGE DETAIL	APRSDB
APRSP1	AUTOMATIC MASTER	APRSDB
APRSQLF	IMAGE DETAIL	APRSDB
P1-OFF-NO-IMS	INDEX (KSAM)	

Now viewing Page 1 of Page Width 1
Press any key to continue...


```
   DATA SET INQUIRED: JOPLAN
   DESCRIPTION:       JOB ORDER PLAN DATA SET
  :-------------------------------------------------------------------+
  : SOURCE OF DATA:  CTRMR                         LOCATION: PUB      :
  :                  CONTRACT RATE                                    :
  :                                                                   :
  : PROGRAM NAME:    MEUPDT                                           :
  : TITLE: UPDATE JOPLAN HOURS                                        :
  +-------------------------------------------------------------------:
  :-------------------------------------------------------------------+
  : SOURCE OF DATA:  JOPUD33                       LOCATION:          :
  :                  ACTUAL ON JOB NUMBER                             :
  :                                                                   :
  : PROGRAM NAME:    JOPUPD                                           :
  : TITLE: UPDATE JOPLAN                                              :
  +-------------------------------------------------------------------:

End of Page
Press any key to continue...
  :-------------------------------------------------------------------+
  : SOURCE OF DATA:  TRNJPRTE                      LOCATION: PUB      :
  :                  MERGE FILE TO UPDATE JOPLAN                      :
  :                                                                   :
  : PROGRAM NAME:    JOPRTE                                           :
  : TITLE: UPDATE RATE ON JOPLAN                                      :
  +-------------------------------------------------------------------:
```

LPO PREPROCESS



PN - 4/5/90

TITLE: Data Synchronization

AUTHOR: Lawrence Kingsbury
Gateway Systems Corp.
2400 Science Parkway
Okemos, MI 48864
517-349-7740

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 4038

Designing Distributive Systems for PC Networks

by
Paul G. Mole'

PMC, Inc.
45 Brandt Avenue, Unit 2
Clark, NJ 07066

(201) 381-5400

Most old-time HP3000 professionals first gained their application experience from working in large IBM shops. By going through a hardware transition to the HP3000 computer, most discovered the flexibility and efficiency not found with applications on mainframe computers. Skills were no longer departmentalized to expert sets, such as an CISC expert or a IMS expert. In the HP3000 environment most of us learned quickly to be the screen, database, and operating system expert. With the ease of use the HP3000 professional became familiar with three types of application design - centralized, decentralized, and distributive.

Although the HP3000 is adept at handling all three designs the current trend toward using personal computers for applications platforms necessitates another hardware transition for the HP3000 professional. Like typical systems developed on the HP3000, personal computers allow data to be distributed across an organization and encourage system ownership by the using departments. They are fast cost-effective tools which place more CPU power on the user's desk.

The design of distributed applications can be complex, but they come the closest to balancing performance with application control. Like the HP3000 environment, these systems have centralized backups, network management and administration. However the data and application programs are kept where they are most needed, to ensure high performance and faster response times.

These distributed systems require some intelligence and data storage capabilities at various points in the design, which can drive costs up. Likewise, because of the mix of software and hardware needed, custom software usually must be used to remedy the incompatibilities.

Still when developing a distributed application, the approach parallels development on the HP3000. First a person must be selected to be responsible for the data, its structure, its transactions, its ownership, and general operations - namely the System Administrator. The System Administrator would be responsible for tuning the application's performance, evaluating problems, insuring the integrity of the data and the system. This person would develop operational procedures for backups, disaster recovery plans, and other similar standards found for most HP3000 applications.

The distributive applications would go through the same project life cycle as other software applications - starting with the Systems Requirements Definition Stage. This stage ensures that all the requirements of the application are known and understood. But with distributed design this stage must also address:

- o Which data should be unique to a work group and which data should be shared.
- o How to group the components of the application.

The idea of sharing and localizing data is similar to the security and design for Image databases. Data unique to a work group is "local" data used by a workstation, file server, or a host computer for a department or user area. For example, at a particular sales office, the personnel information concerning the salesman at that office could remain local; and be maintained independently by that sales office. The particular sales office could decide how to format local information, such as birthdates, hired dates, names, etc. This does drive-up the demand to develop custom programs to manipulate and report on this data.

Shared data is available to more than one work group, and resides on a centralized CPU, like the HP3000 or a file server. Using the sales office example, the salesman's ID number with his sales statistics could be categorized as shared data. Therefore, the format of the salesman's ID number would be standardized across the application.

The second issue to keep in mind during the definition stage is grouping the individual functions, or components, into design modules. Each Module would contain the set of operations it needs to work. Again like HP3000 applications, the design must group a mix bag of individual needs into cohesive logical modules so that they can eventually be developed into programs.

Also, at this stage the control of transactions, security, audit trail, and other constraints are determined for both local and shared data. File formats, edit validations, algorithms are defined not to form the foundation of the application.

The one main difference that the HP3000 professional will find with designing distributed application is that the data and programs can exist in various locations depending on the needs of the application. This presents an opportunity when designing the application. For example, if the design limits the need for shared data then the local end users would be able to continue to operate when a host computer is down. This has a significant advantage over centralized HP3000 applications.

Two key design issues must be remembered for distributive applications:

o Concurrent control
o Data integrity.

Concurrent control is similar to an Image dataset locking strategy where you want to control the way requests for data are processed. A "deadly embrace" can occur with distributed systems when one program waits for a resource that is held by another program which is waiting for a resource held by the first program. These resources include memory, CPU, disk space, devices, etc.

Also like Image datasets, programs can lock exclusively resources and deny access to them by other programs. This causes response problems similar to Image locking transactions where a user is waiting for another user to complete their transaction. If someone does not complete the transaction another part of the application remains frozen. Designing in time-outs and conditional locks for critical resource must be considered to protect the application.

Data integrity refers to the same attributes found in HP3000 applications - accuracy, security, and timeliness. The programs must be designed to ensure all three attributes. The design must take into account the types of access to the data that users require, i. e READ ONLY vs. READ/WRITE.

Different application design tools are available depending on the development approach. Two common development approaches include:

o Client/Server Model
o SQL Relational Tools

The Client/Server Model approach generally uses programs on the end user's personal computer (the Client) connected to a LAN database management system (the Server), which retrieves data, processes requests, and returns it to the user's computer. This approach has basically three designs:

1. Intelligent Clients/Dumb Servers - Here all the processing of data is performed by the intelligent clients and the dumb servers perform only input and output functions.

2. Dumb Servers/Intelligent Servers - With this model the reverse is true to take advantage of the efficiency of a large host processor, like the HP3000.
3. Intelligent Clients/Intelligent Servers - Both machines process the data for efficient use of resources.

The second development approach is to use SQL Relational Tool Products to access, retrieve, and manipulate data. SQL is an industry standard non-procedural language designed for relational databases. It is made up of the following components:

1. Data Definition Language
2. Data Control Language
3. Data Manipulation Language

The Data Definition Language is responsible for the database structure (schema). Field attributes, file relations, edits, descriptions, etc are handled by the Data Definition Language. The Data Control Language handles the multi-user aspects of the application. Lasting the Data Manipulation Language provides for adding, modifying, and querying data.

Because SQL has become an industry standard, it is available for a wide variety of hardware platforms and network operating systems. SQL is incorporated in various products including programming languages, fourth-generation tools, and spread sheets. Although SQL is resource intensive, it is becoming the preferred approach for developing distributed applications.

Along with SQL several vendors offer distributed databases, which run across multiple platforms and provide a wide range of compatibility. These distributed databases normally provide a fourth-generation development environment and support various programming languages, such as C, COBOL, and FORTRAN. These products include Oracle, Progress, Ingres to name a few.
Most vendors also have a trial package for a first-hand evaluation of their product. These vendors provide proprietary front end packages, which can greatly reduce the development cycle. For example, Progress offers the following proprietary packages:

o Progress 4GL/RDBMS - a central server for automatic concurrency, recovery, and distributive database functions.
o Progress Fast Track - a 4GL language, menu editor, screen painter, report writer, query by form generator.
o Progress Query/Report - query/report generation tool

With these types of packages available for distributive applications, the development time is greatly reduced. The complexity of recoveries and lock strategies are handled by these types of products. Cross-platform communication, even cross-platform development, can be handled by these types of products.

Another tool for distributed applications development is called "database servers", which are similar to the client/server model. Here a server, or a "back end", resides on the network server and processes requests for data. The client side, or "front end", resides on the user's personal computer. Its function is to retrieve and present the data in various complex formats. Vendors, such as Ashton-Tate/Microsoft, Sybase, and Novell, offers these types of database servers.

Other development tools for distributive applications development include products called programmers workbenches. These products include an editor, compiler, debugger, and other programs necessary for the development of software. One tool in this category, called remote procedure call (RPC), provides interprocess communication between software running on two different computers. With this tool an application can be split apart with parts executed locally while other parts are executed remotely. The application will use the RPC extensions as a preprocessor when executing.

The number of workbench tools is steadily increasing as the need for interprocessing and distributive software develops. The demand for more choices is still growing, as in the early days of HP3000 computer. Similarly while the MIS department waits for emerging tools and technology , the same types of problems - security, data integrity, and control are again being addressed. Modifications to existing standards for testing, installation, and other development stages must now be incorporated for distributed systems. The tools and techniques may be new, but the control and process are still the same.

The new age of distributive applications has its feet firmly planted in the development methods of the past. Even with new technology, past practical methods of design hold true. Therefore, understanding the process is as important as understanding the new tools and techniques.

Developing Distributed Applications using LM/X

Grant Sidwall
Hewlett-Packard (Canada) Ltd.
1825 Inkster Blvd.
Winnipeg MB Canada R2X 1R3

Introduction

In general terms, a distributed application is one in which the processing required to do the job is distributed across multiple computers. The degree of integration, and the tightness of the connection between the computers involved may vary widely, and yet still fit this definition. The complexity of the design and development process is generally proportional to the degree of integration, i.e., as the separate processes and processors which make up the application become more dependent on each other, the task of creating the application becomes harder.

For example, a remote batch application could be considered to be a distributed application. In such a system, the data required is often entered, editted, and verified on a small 'local' computer. The data may then be stored onto a portable medium, such as tape or diskette, and carried or even mailed to another, usually larger, computer where the data is processed to achieve the desired results. In this case, there is very little integration between the computers used. The only requirement is that the larger 'batch' machine be able to read the media produced by the data entry machine. To the system developer, this means that the only common knowledge required between the two machines is the format of the data on the diskette or tape. There is no need to co-ordinate the initiation of the two 'ends' of the application in real time, and the only protocol for passing messages between them is the human handling of the intervening storage media. So, with regard to integration, the developer's task is:
— define a format for the intervening media.

Providing a communication link between the two systems would increase the degree of integration in that it would avoid the 'human intervention' required for the transport of data between the machines. This addition makes it necessary to define a protocol between the machines to ensure that the data sent is received without error. Also, timing constraints are now placed on the operation; the data entry machine must transmit the data only when the large processor is expecting it. This means that the developer must now be concerned with the establishment of the logical (and maybe physical) connection between the machines, and with the method by which the data will be transferred. She/he must now decide which end will initiate the establishment of the connection, and make sure that the other end is ready to accept it at that time. Then a protocol for transmitting the data without error must be designed or chosen, and a method of gracefully terminating the transfer and closing the logical connection must be found. In this case, the work of establishing the connection, making sure that the data transmission is error free, and writing it into a file on the receiving system is usually accomplished by a data communication subsystem. It still makes the development process more complex, since the integration task list is now:
— define or choose a physical communication method.
— define a method for establishing the connection.
— define or choose a protocol for file transfer.

Perhaps the next step in integrating this application would be to have the data fully validated as it is entered to avoid a manual verification step. To accomplish this might require that the database on the large machine be checked for valid values for key fields in the data record as the record is being entered. At this point, the two processes become tightly coupled, since the data entry can no longer proceed without co-ordinated activity on the database machine. The integration tasks have become:
— choose a physical communication method.
— define a method for establishing communication.
— define a message format.

— define a protocol for passing messages in both directions.
— develop both ends of the application to be 'interactive', so that they stay 'in synch'.

This is the type of integrated application addressed by this paper, specifically applications in which the 'large' machine is an HP-UX system, the 'small' machine is a MS-DOS based personal computer, and the connection between them is provided by LAN Manager/X (LM/X). The architecture of such an application is referred to as 'client-server' in that the HP-UX system is acting as a 'server' of data to the 'client' personal computers. This paper will focus on one method of establishing a connection between the systems, and then using that connection to transfer data.

LM/X overview

LAN Manager/X (LM/X) is a networking product designed to integrate personal computers into the HP-UX environment. It provides PC's running MS-DOS or OS/2 with the ability to use file systems, printers, and other resources attached to HP-UX systems in a Local Area Network (LAN) environment. It also provides program developers with an Application Programmer Interface (API) to facilities for communicating between processes on different processors in distributed applications.

The API contains more than 140 function calls that are divided into 26 categories. Covering all of the categories, let alone all of the functions, is beyond the scope of this paper. This paper will focus on one of the categories, namely Named Pipes. Named Pipes provide:
— a defined method for establishing the connection between PC and mini over a LAN.
— an error free protocol for passing messages in both directions.

This leaves the application developer to define a message format and content and provide client and server programs that co-operate with each other. This paper focuses on the basic issues of:
— establishing the connection.
— moving data back and forth over the connection.
— serving multiple clients simultaneously from one server.

Named Pipes

Named Pipes provide communication between processes on PC's and HP-UX systems that is similar in nature to the named pipes facility that provides interprocess communication within a single HP-UX based processor. There are, however, some important differences:

- Named pipes on an HP-UX system provide peer-to-peer communication, i.e., either of the processes involved can initiate the connection. Named Pipes in LM/X are always created on the server (HP-UX) machine before they can be used by the client (PC) processor.
- The named pipes provided in an HP-UX system are created as part of the file system and so may be treated like ordinary files. This means that a named pipe is inherited from parent to child across a fork() call, and that asynchronous I/O mechanisms such as select() may be used on them. On an HP-UX system, LM/X Named Pipes are implemented using shared memory, not as part of the file system, and can not be treated the same way as ordinary named pipes.
- The LM/X Named Pipe may have multiple *instances*, like wires within a cable, as opposed to the single data flow analogy of a pipe implemented in HP-UX. The idea is that multiple clients could connect to separate instances of the same pipe without each having to know a unique pipe name. The pipe name is more easily associated with a type of service or a path to specific data, rather than being thought of as a connection between the server and a specific client. That is, good design would see pipes named for the service they provide to clients, rather than being named for the client to which they connect.

The steps for creating the server side of the connection are:

1. Create the Named Pipe using *DosMakeNmPipe()*. This call creates the Named Pipe and the first *instance* of it.
2. Wait for a client process to connect to the other end of this *instance* of the Named Pipe using *DosConnectNmPipe(). The server process will block at this point until a client process connects to the other end of this instance of the pipe.*
3. Perform data transfers between the server and client using *DosWrite()* and *DosRead()* calls.
4. Disconnect from the current instance of the Named Pipe using *DosDisconnectNmPipe()*.
5. Remove the pipe with a *DosClose()* call.

The steps for creating the client (PC) side of the connection are:

1. Make the connection to the existing instance of the Named Pipe using *DosOpen()*. This call will return ERROR_PIPE_BUSY if there is no available instance of the specified pipe.
2. Perform data transfers between the client and server using *DosRead()* and *DosWrite()* calls.
3. Close the connection with a *DosClose()* call.

To illustrate these steps, I have created a simple, unsophisticated application consisting of an HP-UX based server program *server1.c* and a PC based client program *client1.c*. The purpose of the application is to show, with as little code as possible, the basics of Named Pipes on LM/X. To that end, the error checking is very simple (the programs abort if an error is encountered), and no provision is made for losing the connection and then re-establishing it (such as when a user cycles power on a PC).

The server program will run until it is interrupted by the user sending it an interrupt signal (usually by typing control C). The client program runs to completion each time it is invoked, and expects its one parameter to be given on the command line. The function of the application is to take a user name from the command line of the client process on the PC, transmit it to the server process on the HP-UX machine, search the /etc/passwd file for it (just using *grep*), and send a message back to the client PC for display. A detailed explanation of the most vital lines of each program follows. Refer to the server program listing on pages 6 and 7, and the client program listing on page 8 to see the extracted lines in context.

Simple Server Program : server1.c

1. *static unsigned short phandle;* - the handle by which the Named Pipe is known throughout the program is declared globally. This has to do with the fact that Named Pipes are implemented using shared memory, and as such the shared memory resources are not returned to the operating system automatically when the program is terminated 'abnormally'. Since the program is written to run forever, it can only be terminated by sending it a signal. This signal (assumed to be SIGINT, the interrupt signal) is trapped, and the interrupt handler *intquit* is invoked. The only way the interrupt handler can have access to the Named Pipe's handle is if it is a static global.
2. *static char pipename[] = "\\pipe\\userpipe";* - since the C language uses the backslash as a special character quoting mechanism,the pipe name must use double backslashes to separate it's components, though single backslashes are all that appear in the resource name. The first component must be the word *pipe*, and the second must be a pipe name that is unique on this machine.
3. *sigset (SIGINT, intquit);* - set up the interrupt handling routine *intquit()* to handle the interrupt signal *SIGINT*.

4. *omode = 2;* - bits 0-2 of the open mode parameter determine the permitted direction(s) of data movement through the pipe. A value of 2 means full duplex, i.e., data can flow in both directions - client to server and server to client. A value of 0 means outbound (from server to client) only. A value of 1 means inbound (from client to server) only. Bit 7 controls whether spawned processes can inherit the named pipe's handle across an *LMForkNmPipe()* call. A value of 0 means inheritance is permitted.

5. *pmode = 0x0500 | 0x0001;* - the pipe mode parameter sets the operating characteristics for the pipe. Bit 14 (0x8000) sets the blocking mode - if 0, all reads block if no data is available, and writes block until all data in the buffer is written. If 1, reads and writes return immediately if no data can be moved (pipe full on a write or empty on a read).
   Bit 10 (0x0400) determines whether the pipe is in byte (=0) or message (=1) mode. In message mode, a read at the receiving end will, if the receiving buffer is large enough, block until all the data in the transmitted message is available, i.e., the entire message is read at once. In byte mode, a read returns whatever data is available, without regard to message boundaries.
   Bit 8 (0x0100) determines the pipe read mode. If it is 0, the pipe can only be read as a byte stream. If it is 1, the pipe can be read in byte stream or message mode.
   Bits 0-7 (0x00FF) determine the maximum instance count possible for this pipe. This is the maximum number of connections between the server and client PC's that can exist simultaneously. The value 0xFF allows unlimited instances, and the value of 0 can't be used.

6. *timeout=0;* - sets the number of milliseconds a PC client that has been unable to connect to an instance of the pipe will wait if it calls *DosWaitNmPipe()* to wait for the next available instance (default is 50 milliseconds).

7. *err = DosMakeNmPipe (pipename,&phandle,omode,pmode,szin,szout,timeout);* - the *address* of the pipe handle is passed in since the value of the pipe handle is returned by this call.

8. *err = DosConnectNmPipe (phandle);* - the pipe exists, and now we wait for a client to open the other end of this instance of it. The server will block here until a PC client opens the other end of the instance with a *DosOpen()* call. This program is very simple, and is based on the assumption that it will only serve one client at a time i.e., only one client request at a time will be in process. When that request is done, we close the connection with that client, and come back to this point to wait for the next client to call *DosOpen()*. Obviously, this could be a problem if there are many clients, and/or it takes significant time to service each request. More on this topic later in the 'enhancements' section.

9. *err = DosRead (phandle,buf,szin,&log);* - a client PC has connected to the other end of this instance of the pipe, and, according to our design of the application, we wait here for it to send data (the user name). If it was also waiting for us to send data, the application would lock up. (Hence the reference to the *design* of the application). Note that *log* returns the number of bytes actually read, so it must be passed as an address. Passing it as a direct reference usually results in a core dump.

10. *retcode = system (sysbuf);* - call *grep* to search */etc/passwd* for any occurrence of the user name passed in (not a thorough check for a valid user id!). Output is redirected to */dev/null* to avoid having the results of successful searches printed on the HP-UX user's terminal or system console.

11. *err = DosWrite (phandle,buf,strlen(buf)+1,&log);* - it's a good idea to include the terminating null in the transmitted ASCII string for ease of use at the receiving end. Note that *log* returns the number of bytes sent, and so must be passed as a pointer here as well.

12. *err = DosDisconnectNmPipe (phandle);* - we have finished with this client. Remove the pipe instance connected to it, so the instance can be used by another client, or a new invocation from the same client. Then back to waiting for another client request.

13. *err = DosDisconnectNmPipe (phandle); (in routine intquit())* - the interrupt could come at any point in the process, so disconnect before closing.

14. *err = DosClose (phandle);* - we are through with this pipe, so deallocate the resources it was using. No further connections are possible until another *DosMakeNmPipe()* is executed.

Simple Client Program : client1.c

1. *#include <os2.h>* - and associated *#define's*. The documentation assumes that the client application will be built on an OS/2 workstation and then bound to the dynamically linked libraries using the OS/2 *BIND* command to create an application that can be run on DOS. This program was developed only on a DOS station (more on that later under **Implementation Considerations**), but it does make sense to be including OS/2 headers. Otherwise the *LINK* will yield undefined externals of the form *_DosWrite* (the libraries have all uppercase external names).
2. *static char pipename[] = "\\\\\\\\hpuwcad\\\\pipe\\\\userpipe";* - LM/X shared resources have the form **\\\\servername\\resource** and the C compiler uses the backslash character as an escape character, so we need twice as many. We have to specify the system where the pipe to which we are connecting exists, unlike the server which is creating the pipe on the system where it is running.
3. *attribute = 0;* - the pipe may have the attribute of being read-only (attribute=1), or normal (read/write) if attribute=0.
4. *oflag = 0x0001;* - specifies the action to take if the pipe exists. If oflag=0x0000 and the pipe exists, the open will fail. If oflag=0x0001, the pipe is opened if it exists (which is what is wanted here).
5. *omode = 0x0002 | 0x0010;* - the open mode gives the characteristics of the open pipe.
   Bits 0-2 set the direction(s) of data flow - 0 implies inbound, from the client to the server (outbound in the local sense). 1 implies the reverse and 2 implies a bidirectional pipe.
   Bits 4-6 are documented as reserved. In the OS/2 Programmer's Reference, they are specified as being used to control the sharing mode of the file being opened. Since instances of pipes are not shared, this would seem to be unnecessary, but it turns out that a form of sharing must be specified. A value of 0x40 (open access) or 0x10 (exclusive access) works.
   Bit 7 sets inheritance of the pipehandle - 0 permits inheritance.
6. *err = DosOpen (pipename,&phandle,&action,0L,attribute,oflag,omode,0L);* - the open call returns the pipe handle at the client end, and the 'action' taken when the open was attempted - any returned value other than 1 is a failure of the open.
7. *err = DosWrite (phandle,cmd,strlen(cmd)+1,&log);* - very similar to the server write call.
8. *err = DosRead (phandle,buf,BUFSIZE,&log);* - same again, just get the order right so the client is reading while the server is writing, and vice versa.
9. *DosClose (phandle);* - close this end of the instance of the pipe. The client is finished.

Listing 1 - server1.c

```
/*  server1.c - simple LM/X user name verification server */

#include <signal.h>
static unsigned short phandle; /* global for interrupt handler */

main()
{
/* parameters for 'system' call */
char sysbuf[80];
int retcode;

/* parameters for Named Pipe */
#define BUFSIZE 100
static char pipename[] = "\\pipe\\userpipe";
char buf[BUFSIZE];
unsigned short omode, pmode, szin, szout, err, log;
long tmout;

/* set up proper termination on interrrupt (control C) signal */
int intquit();
sigset (SIGINT,intquit);

/* create an instance of the Named Pipe */
omode = 2;                          /* bidirectional */
pmode = 0x0500 | 0x0001;  /* message mode, one instance */
szout = BUFSIZE;          /* outgoing buffer size */
szin = BUFSIZE;                     /* incoming buffer size */
tmout = 0;                          /* use default */
err = DosMakeNmPipe (pipename,&phandle,omode,pmode,szin,szout,tmout);
if ( err != 0 ) quiterror ( "DosMakeNmPipe",err);

/* This is the loop that serves until interrupt signal */
while (1) {

  /* Connect this instance of the pipe to a client at its DosOpen()*/
  err = DosConnectNmPipe (phandle);
  if ( err != 0 ) quiterror ( "DosConnectNmPipe",err);

  /* Read a user name from the client */
  err = DosRead (phandle,buf,szin,&log);
  if ( err != 0 ) quiterror ( "DosRead",err);

  /* Use grep to see if user name exists in /etc/passwd */
  sprintf (sysbuf,"%s %s %s","grep",buf,"/etc/passwd > /dev/null");
  retcode = system (sysbuf);

  /* Place appropriate message in buffer */
  if ( retcode == 0 ) {
    /* user exists */
    strcpy (buf,"User exists on this system");
    }
  else {
    /* user not found */
    strcpy (buf,"User does not exist on this system");
  }
```

```
  /* Send message back through pipe */
  err = DosWrite (phandle,buf,strlen(buf)+1,&log);
  if ( err != 0 ) quiterror ("DosWrite",err);

  /* Disconnect from this instance of pipe */
  err = DosDisconnectNmPipe (phandle);
  if ( err != 0 ) quiterror ("DosDisconnectNmPipe",err);

  /* Now go back and reconnect to pipe and wait for next client */
  }
}


intquit()
{
int err;
/* Disconnect from the Named Pipe and close (deallocate it)
   so that successive invocations don't eat up shared memory */
err = DosDisconnectNmPipe (phandle);
if ( err != 0 ) quiterror ("DosDisconnectNmPipe",err);
err = DosClose (phandle);
if ( err != 0 ) quiterror ("DosClose",err);
printf ("server1 terminated\n");
exit (0);
}

quiterror (string,err)
char *string;
unsigned short err;
{
printf ("%s failed : error %d\n",string,err);
exit (1);
}
```

Computer Museum

Listing 2 - client1.c

```
/*  client1.c - simple LM/X user name verification client */
#define INCL_DOSERRORS
#define INCL_DOS
#include <os2.h>

main(argc,argv)
int argc;
char *argv[];
{
/* parameters for Named Pipe */
#define BUFSIZE 100
static char pipename[] = "\\\\hpuwcaa\\pipe\\userpipe";
char buf[BUFSIZE];
char *cmd;
unsigned short phandle, action, omode, oflag, err, log;
unsigned int attribute;

/* get the user name to be validated from the command line */
if ( argc != 2 ) {
  printf ("Usage : %s username\n",argv[0]);
  exit (1);
  }
cmd = argv[1];

/* Open our end of the Named Pipe */
attribute = 0;                   /* normal (not read-only) */
oflag = 0x0001;                  /* open the pipe if it exists */
omode = 0x0002 | 0x0010;  /* bidirectional + share mode bits */
err = DosOpen (pipename,&phandle,&action,0L,attribute,oflag,omode,0L);
if ( err != 0 ) quiterror ( "DosOpen",err);

/* Send user name through pipe to server */
err = DosWrite (phandle,cmd,strlen(cmd)+1,&log);
if ( err != 0 ) quiterror ("DosWrite",err);

/* Read message from server and print to user */
err = DosRead (phandle,buf,BUFSIZE,&log);
if ( err != 0 ) quiterror ( "DosRead",err);
printf ( "%s\n",buf);

/* Close the pipe */
DosClose (phandle);
}

quiterror (string,err)
char *string;
unsigned short err;
{ printf ("%s failed : error %d\n",string,err);
  exit (1); }
```

Enhancements

The example application works, but it is quite limited. The server can only handle one client at a time, and it disconnects from the pipe to the client immediately after writing to it. On a busy system with multiple clients and moderate network traffic, it would be inadequate. The brute force method of handling multiple clients could be to simply start up more servers, but that is unworkable for at least two reasons. One, the number of servers to be started might never be right if the number of clients varies, and two, the second and all succeeding servers will find that the Named Pipe is busy. The answer then, would seem to be to have the server spawn off multiple children. Each child would handle a specific instance of the Named Pipe and the requests from the client at the other end. The idea of a pipe having multiple instances now makes more sense: the clients all ask for the same Named Pipe, but they each get a different instance. The problem with this is that Named Pipes are not inherited across a *fork()* in HP-UX.

To overcome this, the LM/X API contains a special form of the *fork()* call, known as *LMForkNmPipe()*. This call functions the same as the standard HP-UX call, and additionally provides for inheritance of the Named Pipe handles, so the child process can access the instance of the pipe that the parent has established. Using this call, a server program can sit in a loop, waiting at a *DosConnectNmPipe()* call until a PC client connects to the other end of the pipe instance being used. The server can then spawn a child to handle the client's requests, and go back to waiting on the *DosConnectNmPipe()* call for another client. When it has finished with the PC client's requests, the child process can close the instance, so it will be available for other clients, and terminate.

To avoid closing the pipe instance before the client has read the data in it, the child server could simply wait an arbitrary time, but a more certain method is for it to make a call to *DosBufReset()*. This call does not return to the child server process until all data has been read from the pipe (or about five seconds have elapsed). Doing this guarantees, as much as possible, that the client will receive the data sent to it.

The creation of a child process to handle the client requests requires some other 'enhancements' that are based on the requirements of process handling in HP-UX, rather than facilities or requirements of LM/X.

When an HP-UX process terminates, it notifies its parent process that it has exitted. This notification can be in the form of an exit status byte returned to the parent if the parent has waited for the child to complete. Or, if the parent did not wait, the notification will be in the form of a signal *(SIGCLD)*, that is sent to the parent. In either case, the child process does not complete until this notification has been received. Instead, it remains as a < *defunct* > entry in the system process table. Since the system process table is of fixed size, it would overflow if such entries were allowed to accumulate.

The easiest way to handle these signals from terminated child processes is to accept them and take no action (this is known as *ignoring* them). To accomplish this, a call to a routine called *sigset()* is made at the beginning of the program. This call, with the action parameter set to *SIG_IGN* changes the way the child's *exit()* statement performs. Instead of staying in its process slot until the *SIGCLD* signal is caught, it frees the process slot immediately and is done.

This represents a problem later on in the child process when the child makes the call to *system()* to execute the search of */etc/passwd*. This call actually generates a child process (a shell) that starts up a child process to execute the *grep* command and waits for it to finish so it can return the command's exit status. If the child process returns no status, things get messed up here, and the search always appears to fail, even though the *grep* command runs correctly (it's return value is always -1). To prevent this, a call to *sigset()* is made at the start of the child process, restoring the default action for the receipt of a child termination signal (a child generates the signal, it is accepted, but nothing is done).

Listing 3 of program *server2.c* on pages 10 and 11 shows the changes made. Following the listing is a description of the changes in order of occurrence.

Listing 3 - server2.c

```
/*  server2.c - enhanced LM/X user name verification server */

#include <signal.h>
#include <lm/error.h>
static unsigned short phandle; /* global for interrupt handler */
extern int errno;

main()
{
/* parameters for 'system' call */
char sysbuf[80];
int retcode, pid;
/* parameters for Named Pipe */
#define BUFSIZE 100
static char pipename[] = "\\pipe\\userpipe";
char buf[BUFSIZE];
unsigned short omode, pmode, szin, szout, err, log;
long tmout;

/* set up proper termination on interrrupt (control C) signal */
int intquit();
sigset (SIGINT,intquit);
/* *** ENHANCEMENT *** catch & ignore death of child signal */
sigset (SIGCLD,SIG_IGN);

/* set values to create an instance of the Named Pipe */
omode = 2;                          /* bidirectional */
/*  *** ENHANCEMENT *** serve up to 32 clients at once */
pmode = 0x0500 | 0x0020;  /* message mode, 32 instances */
szout = BUFSIZE;          /* outgoing buffer size */
szin = BUFSIZE;           /* incoming buffer size */
tmout = 0;                          /* use default */
/* *** ENHANCEMENT *** include creating the instance in the loop */
/* This is the loop that serves until interrupt signal */
while (1) {
  err = DosMakeNmPipe (pipename,&phandle,omode,pmode,szin,szout,tmout);
  if ( err != 0 ) quiterror ( "DosMakeNmPipe",err);

  /* Connect this instance of the pipe to a client at its DosOpen()*/
  err = DosConnectNmPipe (phandle);
  if ( err != 0 ) quiterror ( "DosConnectNmPipe",err);

  /*  *** ENHANCEMENT *** start up a child process to do the work */
  pid = LMForkNmPipe();
  if ( pid == -1 ) quiterror ("LMForkNmPipe",errno);

  if ( pid == 0 ) {
    /* *** ENHANCEMENT *** restore normal processing of SIGCLD */
    sigset (SIGCLD,SIG_DFL);

    /* Read a user name from the client */
    err = DosRead (phandle,buf,szin,&log);
    if ( err != 0 ) quiterror ( "DosRead",err);

    /* Use grep to see if user name exists in /etc/passwd */
    sprintf (sysbuf,"%s %s %s","grep",buf,"/etc/passwd > /dev/null");
    retcode = system (sysbuf);
```

```
    /* Place appropriate message in buffer */
    if ( retcode == 0 ) {
      /* user exists */
      strcpy (buf,"User exists on this system");
      }
    else {
      /* user not found */
      strcpy (buf,"User does not exist on this system");
    }

    /* Send message back through pipe */
    err = DosWrite (phandle,buf,strlen(buf)+1,&log);
    if ( err != 0 ) quiterror ("DosWrite",err);

    /* *** ENHANCEMENT *** wait for client to empty pipe */
    err = DosBufReset (phandle);
    /* but it's OK if the client has closed it and gone */
    if ( err != 0 && err != ERROR_BROKEN_PIPE )
      quiterror ("DosBufReset",err);

    /* Disconnect from this instance of pipe */
    err = DosDisconnectNmPipe (phandle);
    if ( err != 0 ) quiterror ("DosDisconnectNmPipe",err);

    /* *** ENHANCEMENT *** close this instance  and exit */
    err = DosClose (phandle);
    if ( err != 0 ) quiterror ("DosClose",err);
    exit (0);
    }

  /* Close this connection (child is using it) and get new one */
  err = DosClose (phandle);
  if ( err != 0 ) quiterror ("DosClose",err);
  }
}

intquit()
{
int err;
/* Disconnect from the Named Pipe and close (deallocate it)
   so that successive invocations don't eat up shared memory */
err = DosDisconnectNmPipe (phandle);
if ( err != 0 ) quiterror ("DosDisconnectNmPipe",err);
err = DosClose (phandle);
if ( err != 0 ) quiterror ("DosClose",err);
printf ("server2 terminated\n");
exit (0);
}

quiterror (string,err)
char *string;
unsigned short err;
{
printf ("%s failed : error %d\n",string,err);
exit (1);
}
```

Server code enhancements

1. *sigset (SIGCLD,SIG_IGN);* - this call establishes that the parent is not interested in the return status of child processes, so the children terminate without trying to send any status.
2. *pmode = 0x0500 | 0x0020;* - this parameter is now set to allow up to 32 instances of the Named Pipe, not just one.
3. *while (1)* { - when the server dealt with only one client at a time, it used only one instance of the Named Pipe, and so had to create only one. This one instance was reused for each requested connection. Now, we need multiple instances of the Named Pipe, so we must make repeated calls to *DosMakeNmPipe()* to create them. That is why this call is now included in the loop.
4. *pid = LMForkNmPipe();* - this call creates the child process (a copy of the parent process, i.e., running the same code) to handle the server request on the Named Pipe connection just made.
5. *if (pid == 0)* { - the parent and child differ in this value: the value returned to the parent is the new process id of the child, while the value returned to the child is 0. All the code inside this *if* statement is executed *only* in the child process.
6. *sigset (SIGCLD,SIG_DFL);* - all children of this process (and their children) will require notification of the termination of the child, though no specific action will be taken when this happens. Note that this changes this action only for this new (child) process and its descendants, not for the parent process.
7. *err = DosBufReset (phandle);* - wait until all data is read from the pipe before proceeding. This assures that the pipe will not be disconnected from the client while it still contains data. On the other hand, if the client has read its data and closed *its* end of the pipe, this call fails with *ERROR_BROKEN_PIPE*, which is acceptable.
8. *err = DosClose (phandle);* - the Named Pipe instance is closed (destroyed) because we are in the child process which is about to terminate, and there is no way to return the handle and its associated resources to the parent. The parent, in fact, closes this instance as well, since it can't use it while the child is actively using it.

Client code enhancements

The possibility of multiple clients competing for the server's connections requires one enhancement to the client code, implemented in several statements.

1. *#include <neterr.h>* - this file contains the *#define's* for the symbolic values of the returned error codes, eg. ERROR_PIPE_BUSY.
2. *if (err == ERROR_PIPE_BUSY)* { - there exists an interval in the server code between when the connection for one client is made, and when the next call to *DosConnectNmPipe()* is made. If the client tries to connect during this interval, there will be no instance of the Named Pipe available, and ERROR_PIPE_BUSY will be returned by the *DosOpen()* call.
3. *err = DosWaitNmPipe (pipename,5000L);* - in the above situation, we simply wait (up to 5000 milliseconds) for an instance of the Named Pipe to become available.

See listing 4 of client2.c on page 13 for the enhanced client code.

Listing 4 - client2.c

```
/*  client2.c - enhanced LM/X user name verification client */
#define INCL_DOSERRORS
#define INCL_DOS
#include <neterr.h>    /* ENHANCEMENT for symbolic error checking */
#include <os2.h>

main(argc,argv)
int argc;
char *argv[];
{
/* parameters for Named Pipe */
#define BUFSIZE 100
static char pipename[] = "\\\\hpuwcaa\\pipe\\userpipe";
char buf[BUFSIZE];
char *cmd;
unsigned short phandle, action, omode, oflag, err, log;
unsigned int attribute;

/* get the user name to be validated from the command line */
if ( argc != 2 ) {
  printf ("Usage : %s username\n",argv[0]);
  exit (1);
  }
cmd = argv[1];

/* Open our end of the Named Pipe */
attribute = 0;                      /* normal (not read-only) */
oflag = 0x0001;                     /* open the pipe if it exists */
omode = 0x0002 | 0x0010;  /* bidirectional + share mode bits */
err = DosOpen (pipename,&phandle,&action,0L,attribute,oflag,omode,0L);
/* *** ENHANCEMENT *** wait for up to 5 sec if the pipe is busy */
if ( err == ERROR_PIPE_BUSY ) {
  err = DosWaitNmPipe (pipename,5000L);
  if ( err != 0 ) quiterror ("DosWaitNmPipe",err);
  }
else {
  if ( err != 0 ) quiterror ("DosOpen",err);
  }

/* Send user name through pipe to server */
err = DosWrite (phandle,cmd,strlen(cmd)+1,&log);
if ( err != 0 ) quiterror ("DosWrite",err);

/* Read message from server and print to user */
err = DosRead (phandle,buf,BUFSIZE,&log);
if ( err != 0 ) quiterror ( "DosRead",err);
printf ( "%s\n",buf);

/* Close the pipe */
DosClose (phandle);
}

quiterror (string,err)
char *string;
unsigned short err;
{ printf ("%s failed : error %d\n",string,err);
  exit (1); }
```

Implementation issues

To implement this distributed application, the code must be compiled and linked correctly for the machine on which it will run, and the required resources must be made available.

Server implementation

The server code can be compiled and linked using libraries */usr/lib/liblm.a* and */usr/lib/libV3.a* in one step with a command line such as:

cc -o server server.c -llm -lV3

Because the Named Pipe resource belongs to the Named Pipe user **lmpipe**, server processes must be run by the lmpipe user, or must be either *setuid* or *setgid* to the lmpipe user or group. For example, to have a server program run as a member of the lmpipe group, the superuser must issue the following commands:

chgrp lmpipe serverprog
chmod 2755 serverprog

Additionally, the interprocess communication resource **IPC$** must be made available to clients by issuing the LM/X command:

lm share IPC$ <password>

if the server is operating in *share-level* security. The password is optional. In *user-level* security mode, the IPC$ resource is automatically shared if there is an **lmpipe** user defined.

Client implementation

Because client applications are meant to be built on OS/2, building them on DOS is not a one step process. Microsoft C 5.1 was the compiler used (QuickC 2.0, being targetted only for MS-DOS, lacked one library). Compiling and linking the code was done as two separate steps as follows:

CL /C CLIENT.C

LINK /NOE /NOD CLIENT.OBJ
Run File [CLIENT.EXE]: **cr**
List File [NUL.MAP]: **cr**
Libraries [.LIB]: **SLIBCE +**
Libraries [.LIB]: **API +**
Libraries [.LIB]: **DOSNET**

(User input shown in bold, cr = return key)

To access the Named Pipe resource, the user on the PC must issue a NET USE command of the form:

NET USE \\SERVER\IPC$ **<password>**

The password is required in share-level security if one was specified in the *lm share IPC$* command on the server. In user-level security, the user must be a member of the **lmpipe** group, or listed in the **lmpipe** line in */etc/group*. Otherwise, no access to Named Pipes is permitted.

Summary

This paper has shown the basics of using the Named Pipes Application Programmer Interface under LAN Manager/X. If we refer back to the list of integration tasks faced by the developer of an interactive distributed application, we can see how well the Named Pipes of LM/X provide the required tools:

— choose a physical communication method.
LM/X is built on industry standard protocols that are independent of the underlying physical technology. In other words, the application developer need only ascertain that the physical link supports these protocols, and then proceed to design and code without regard to the physical connection.

— define a method for establishing communication.
The method of creating a connection between client and server is well defined, and has tools to allow each end to wait for the other to arrive at the 'connected' state.

— define a message format.
Named Pipes may operate in message mode, which provides transmission of complete messages, or in byte stream mode, which may be better suited to large data transfers between clients having different buffering capabilities. In either case, the developer must still decide what to put in the message, and how to interpret it at the other end. Other members of the LM/X API family provide more 'defined' services.

— define a protocol for passing messages in both directions.
The simple *DosRead()* and *DosWrite()* calls provide a straightforward way of communicating in both directions over the same instance of a Named Pipe.

— develop both ends of the application to be 'interactive', so that they stay 'in synch'.
This is still something the developer must give some attention, but LM/X provides facilities for waiting until the other end of the pipe is in the required state. This means that the application must stay in sequence, but that timing considerations or 'race' conditions can be handled easily.

In summary, LM/X provides a good set of tools for developing applications which span the dissimilar worlds of DOS based personal computers and HP-UX based workstations and minicomputers.

David Robinson
PowerSpec International
403 Cross Lake Dr.
Fuguay-Varina, NC 27526

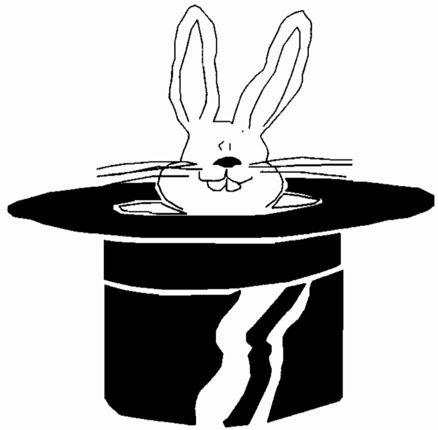
Developing Portable Applications
PowerHouse PC

Paper #4044

POWERSPEC Inc.
The PowerHouse® Specialists

Interex '90

Developing Portable Applications
PowerHouse PC®



®

PowerHouse PC is a registered trademark of Cognos, Inc.
QUIZ, QTP, and QDESIGN are trademarks of Cognos, Inc.

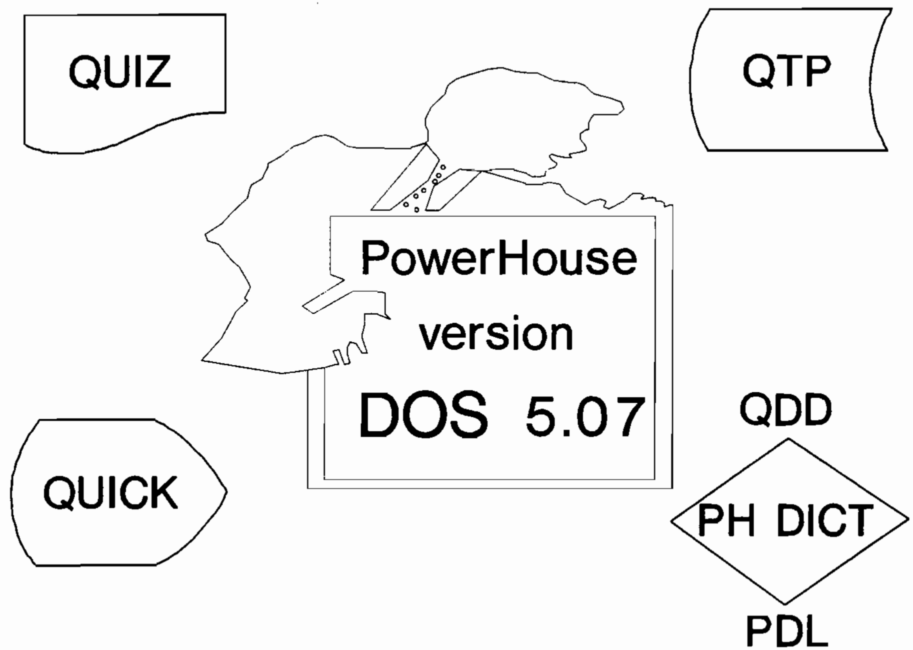
All other marks mentioned are the property of the respective trademark holders

David G. Robinson

PowerSpec Inc.

4044-3

PC PowerHouse



PowerSpec INC. / DEVL PORT APPL / Page 02

Technical Requirements

Hardware

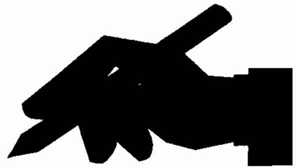
- Computer
  - . IBM PC/AT / Compatible
  - . 286 / 386 PROCESSOR
- Hard Disk
  - . 20 MB
- Monitor
  - . Color or Monochrome
  - . Standard Display Adaptor
- Printer
- Serial Communications Port

Software

- PowerHouse Versions
  - . DOS 640 KB RAM
  - . DOS/EXT 1 MB RAM
  - . OS/2 1.5 MB RAM
  - . LAN 2.5 MB RAM
- DOS Version 3.2 +
- Text Editor

Documentation

- Manuals
  . Getting Started
  . Portability Guide for ...
  . Host Connection for ...
  . New Features
  . Technical Reference

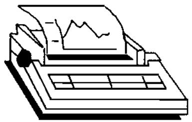


- Installation Kit
  . Installation Guide
  . 7 5.25 Diskettes
  . License Agreement
  . Support Guide

The Operating Environment

VECTRA R/S 16

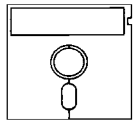
LASERJET II



103 MB Storage

386 Processor
4 MB RAM

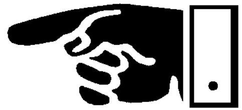
Software



DOS 4.01 D
Desk Top Publishing

The Installing of DOS Version 5.07

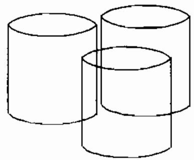
- PC PowerHouse / DOS Version 5.07
- Data Dictionary
  . QDD vs PDL
- DOS Environment
  . CONFIG.SYS
  . AUTOEXEC.BAT
  . PATH
  . BUFFERS / FILES
- Language
  . ENGLISH
  . FRENCH
  . GERMAN
- Editor



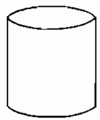
The Installation Process

Subdirectories

PH507
PHDEMO
PHTMP

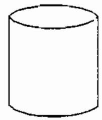


CONFIG.SYS



- ✓ FILES=40
- ✓ BUFFERS=32

AUTOEXEC.BAT



- ✓ SET PHTERM=PCCOLOR
- ✓ SET PHTMP=drive(path)/PHTEMP
- ✓ SET PHVM=drive(path)/PHTEMP
- ✓ SET PHEMS=DISABLE
- ✓ PATH=C:/PH/PH507

Installation Complete

POWERHOUSE
MENU

- Unable to bring up MENU

*** Not Enough Memory ***

- CONFIG.SYS / AUTOEXEC.BAT
  - 1 for POWERHOUSE
  - 1 for DESK TOP PUBLISHING

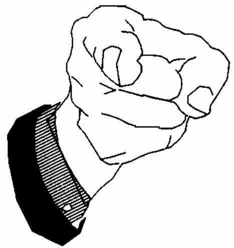
Installation Changes

- Created NEW Files
  - CONFIG.SYS
  - AUTOEXEC.BAT
- Created DOS BAT Files
  - POWERHOUSE
  - DESK TOP PUBLISHING

PWRHOUSE.BAT

```
@ECHO OFF
SET NEEDBOOT=FALSE
COPY C:\CONFIG.SYS       C:\CONFIG.OLD
COPY C:\PH\CONFIG.PH     C:\CONFIG.SYS
COPY C:\AUTOEXEC.BAT     C:\AUTOEXEC.OLD
COPY C:\PH\AUTOEXEC.PH C:\AUTOEXEC.BAT
SET NEEDBOOT=TRUE
IF (%NEEDBOOT%) == (TRUE) WARMBOOT
```

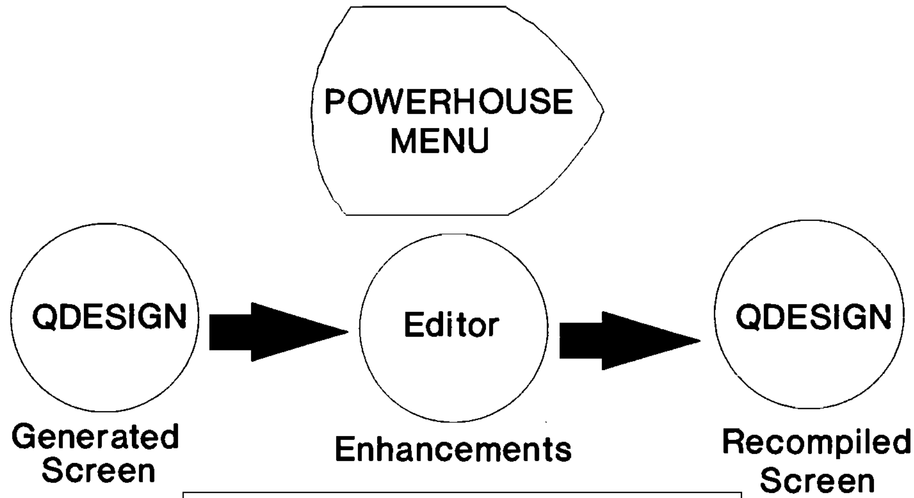
Critique of Installation



- Documentation
  - Poorly Organized
  - Insufficient
    - Installing PC PowerHouse
    - Setting up PC Environment

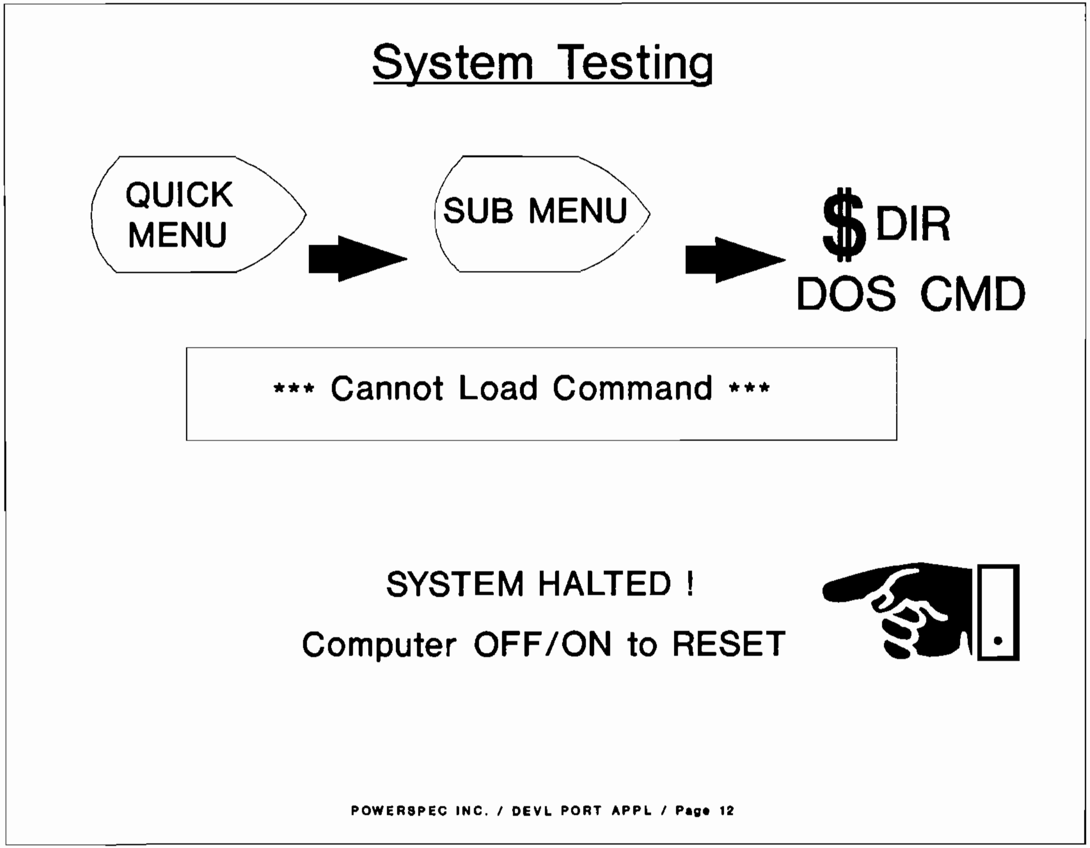
- The Installed System
  - All MESSAGE Files Installed
  - Subdirectory PHDEMO Optional
  - PowerHouse MENU Shell Optional

System Testing



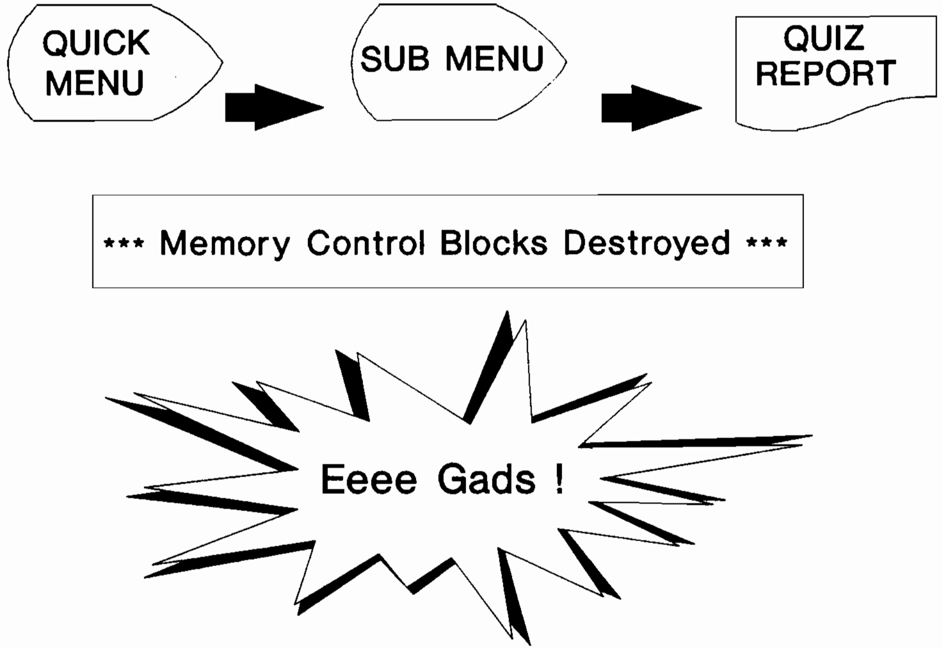
*** not enough memory ***

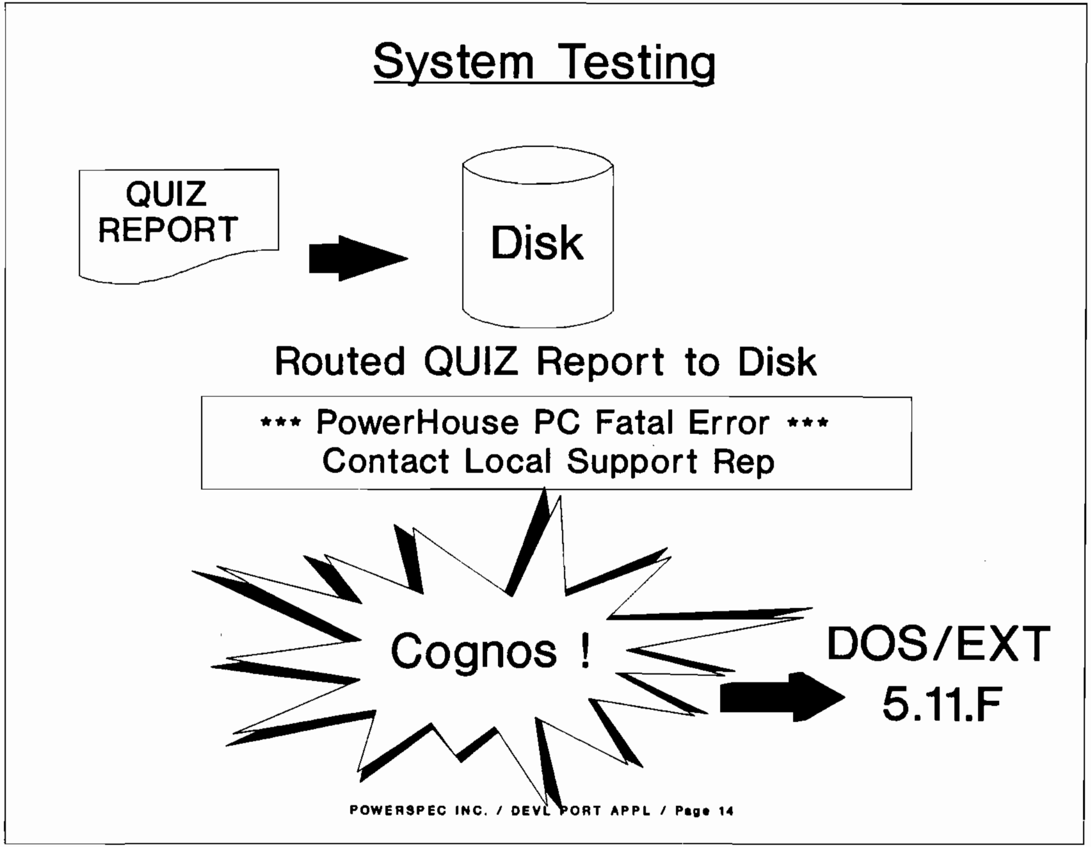
CONFIG.SYS/BUFFERS=20 / REBOOTED !


System Testing
QUICK MENU
SUB MENU
$ DIR
DOS CMD
*** Cannot Load Command ***
SYSTEM HALTED !
Computer OFF/ON to RESET
POWERSPEC INC. / DEVL PORT APPL / Page 12

4044-14

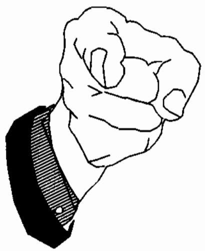
System Testing

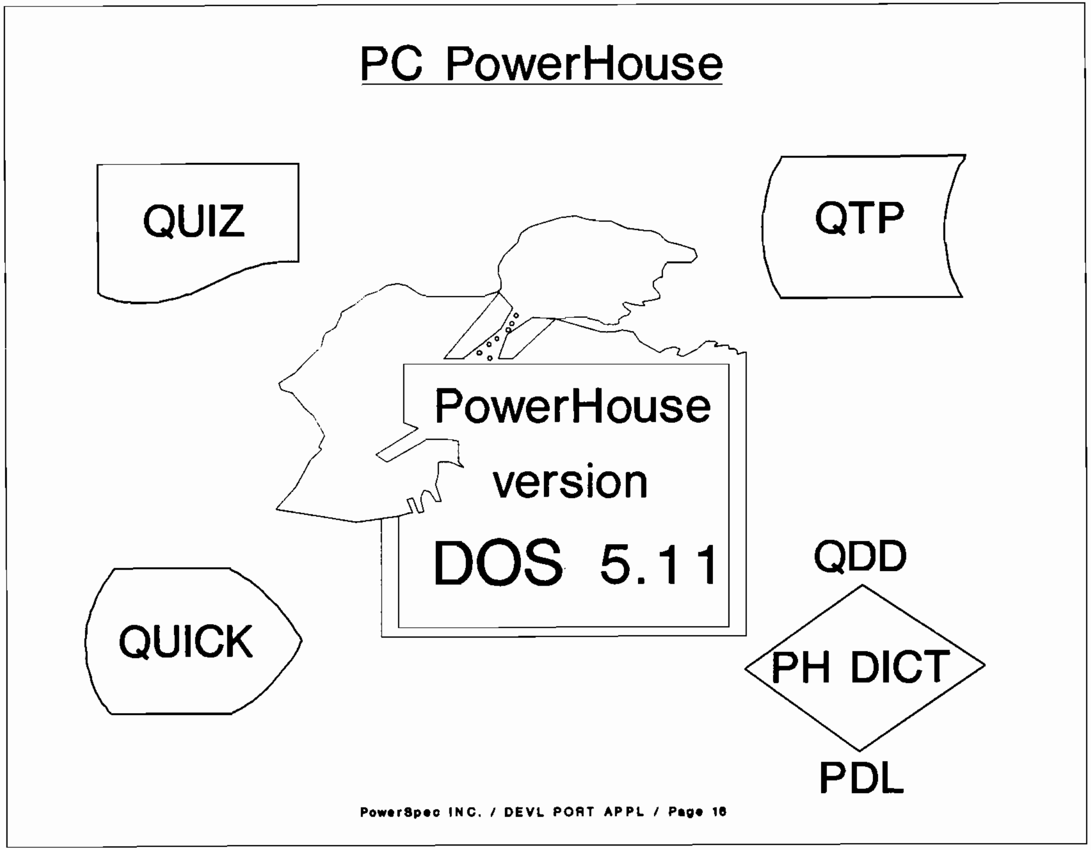



System Testing
QUIZ REPORT
Disk
Routed QUIZ Report to Disk
*** PowerHouse PC Fatal Error ***
Contact Local Support Rep
Cognos !
DOS/EXT
5.11.F
POWERSPEC INC. / DEVL PORT APPL / Page 14

Critique of DOS Version 5.07

- Insufficient Memory
- For Large Scale Development
- Only able to use 640K RAM

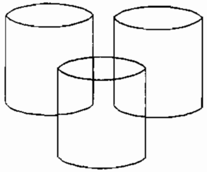



PC PowerHouse
QUIZ
QTP
PowerHouse
version
DOS 5.11
QUICK
QDD
PH DICT
PDL
PowerSpec INC. / DEVL PORT APPL / Page 16

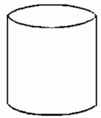
The Installation Process

Subdirectories

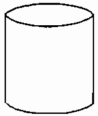
PH511
PHDEMO
PHTMP



AUTOEXEC.BAT



CONFIG.SYS



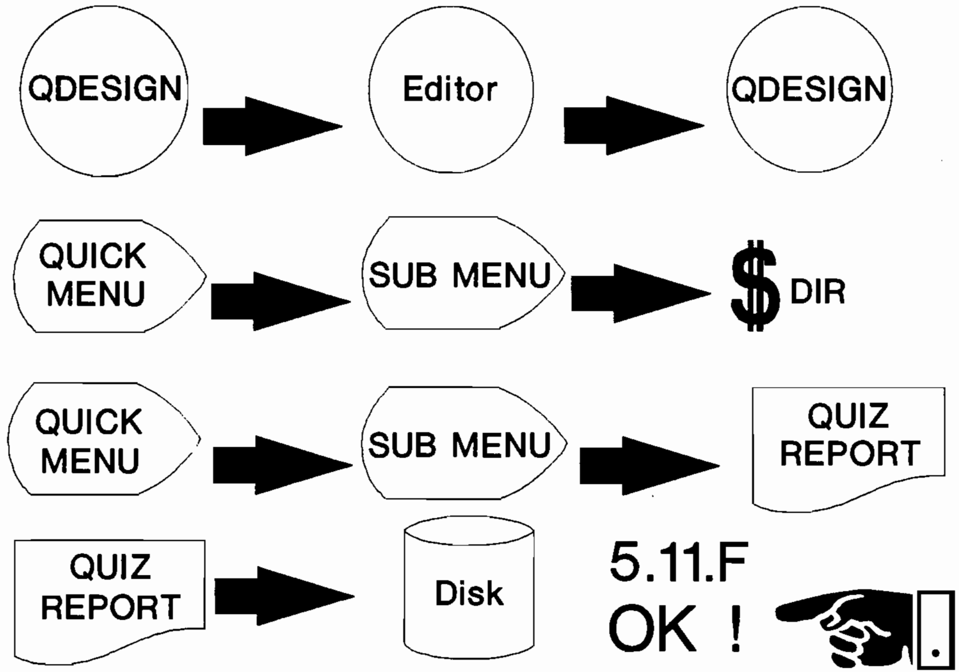
- ✓ FILES=40
- ✓ BUFFERS=20

- ✓ SET PHTERM=PCCOLOR / PCMONO
- ✓ SET PHTMP=drive(path)/PHTEMP
- ✓ SET PHVM=drive(path)/PHTEMP
- ✓ SET PHEMS=DISABLE
- ✓ PATH=C:/PH/PH511
- ✓ SET PHLOCK=OFF

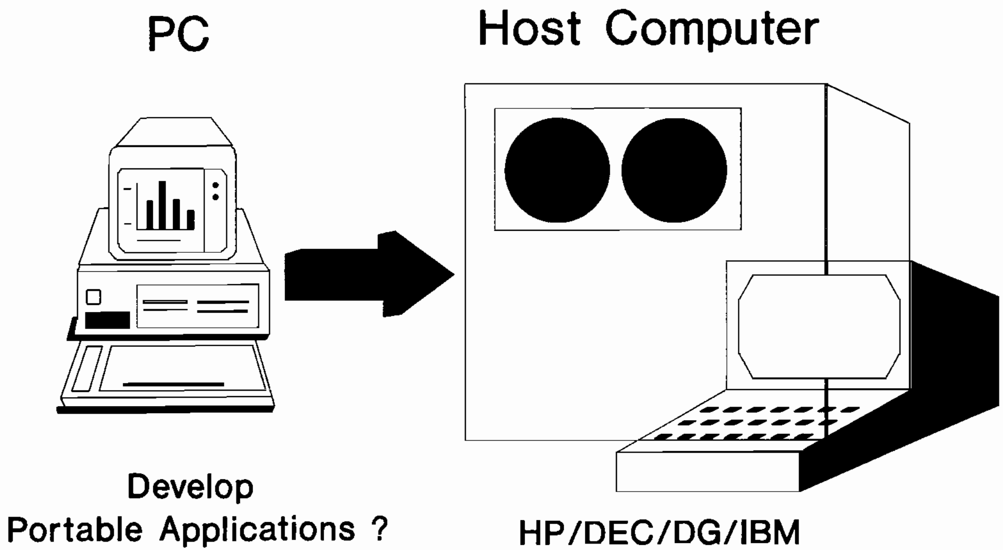
PowerHouse Version DOS/EXT 5.11.F

- DOS/EXT VER 5.11.F
  - . Allows usage of EXTended Memory
- MS/DOS DEVICE DRIVER
  - . HIMEM.SYS = Extended Memory Usage
    - . CONFIG.SYS
      DEVICE = C:/(path)/HIMEM.SYS
- MS/DOS VER 4.01 / HP VECTRAs
  - . HPDCACHE = Disk Caching Program
    - . AUTOEXEC.BAT File
      C:/(path)/HPDCACHE /E+ /S:512

System Testing



Is it Feasible ?



Data Dictionary

QDD

- Hewlett-Packard Users
- Data Structures for PC
  . Sequential
  . Indexed

✓ No Image DBMS
✓ No Relational DBMS

PDL

- DEC VAX Users
- MV/Family AOS/VS Users
- Data Structures for PC
  . Sequential
  . Indexed

✓ No Relational DBMS

Utilities to Convert HOST to PC Dictionary 

Conditional Compilation Codes

- Required to Conditional Execute Code
  - PC a Compilation Code is default

- Constructs

  @ in first position
  @IF
  @ELSEIF
  @ENDIF

- Multiple Compilation Codes Allowed

  @IF HP
  @IF VAX
  @IF DG

- Codes executed during Interpret Phase

  QDESIGN INFO="CC=(HP)"
  QUIZ INFO="CC=(HP,VAX)"

PC to HOST Portable Code

QDD Dictionary Security

```
        QDD INFO="CC=(HP)"
>...
>@IF HP
>USER mgr       LOGONID mgr.sys
>USER class1    LOGONID user1.pwrspec
>@ENDIF
>...
>@IF HP
>FILE clients INDEXED        &
>     READ INCLUDE ALL       &
>     WRITE INCLUDE MGR
>@ENDIF
>...
```

PC to HOST Portable Code

File Sharing

```
   QDESIGN INFO="CC=(HPVAX)"
>...
> FILE clients REFERENCE
>@IF PC
>       OPEN UPDATE
>@ELSEIF HPVAX
>       OPEN 1 UPDATE SHARE
>@ENDIF
>...
```

PC to HOST Portable Code

File Locking/Unlocking

```
   QDESIGN INFO="CC=(HPVAX)"
>@IF HPVAX
>LOCK ctrl-file
>@ENDIF
>GET ctrl-file SEQ OPT
>LET order-no = next-order
>Let next-order = next-order + 1
>PUT ctrl-file
>@IF HPVAX
>UNLOCK
>@ENDIF
>...
```

PC to HOST Portable Code

SCREEN Options

```
QDESIGN INFO="CC=(VAX)"

>@IF HP
>SCREEN x MENU BLOCKMODE  &
>       WINDOW ON LINE 49
>@ELSEIF VAX
>SCREEN x MENU WINDOW ON LINE 49
>@ELSEIF PC
>SCREEN x MENU
>@ENDIF
```

PC to HOST Portable Code

SCREEN HILITEs

```
QDESIGN INFO="CC=(HP)"
>@IF PC
>HILITE DATA        GREEN
>HILITE SCREEN      WHITE ON BLUE
>HILITE ERROR       RED
>@ELSEIF HP
>HILITE DATA        UNDERLINE
>HILITE MESSAGE     INVERSE
>HILITE REQUEST     INVERSE HALFTONE
>@ENDIF
```

PC to HOST Portable Code

Data Structures

QDD INFO="CC=(HP)"

```
>@IF PC
>FILE products   INDEXED
>FILE customers INDEXED
>...

>@ELSEIF HP
>FILE products    IMAGE MASTER OF ...
>FILE customers IMAGE MASTER OF ...
>@ENDIF
```

PC to HOST Portable Code

PC PowerHouse 5.11.F Constraints

✓ QUICK Menu CALLS

```
>@IF PC
>  COMMAND "QUIZ AUTO=report.QZC"
@ELSEIF HPVAX
>   COMMAND "QUIZ AUTO=report"
>...
```

✓ QUIZ Compiled REPORT

✓ By EXTENSION Name

PC to HOST Portable Code

PC PowerHouse 5.11.F Constraints

✓ QUICK Menu CALLS

```
>@IF PC
>  COMMAND "QTP  AUTO=source"
@ELSEIF HPVAX
>   COMMAND "QTP  AUTO=compiled"
>...
```

✓ QTP Compiled RUN

✓ Unable to CALL

PC to HOST Portable Code

PC PowerHouse 5.11.F Constraints

✓ QTP REQUESTs

```
>@IF HPVAX
>RUN backup
>@ENDIF
>REQUEST bkup-file1
>ACCESS file1
>SUBFILE sub1 KEEP INCLUDE file1
>@IF PC
>GO
>@ENDIF
>...
```

✓ Unable to execute multiple REQUEST

E Too Many Open Files

PC to HOST Portable Code

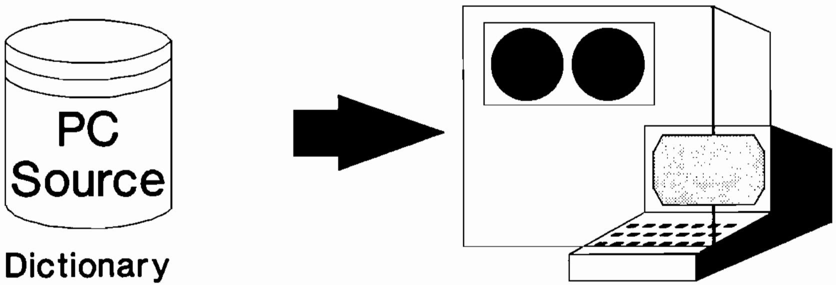
PC PowerHouse 5.11.F Constraints

✓ QUIZ output to DISK

```
>...
>;First Pass
>SET REPORT DEVICE DISK NAME qztotal
>...
>;Second Pass
>@IF PC
>$RENAME qztotal.PRN, qztotal.QZS
>@ENDIF
>USE qztotal
>...
```

✓ Rename File EXTENSION

PC to HOST



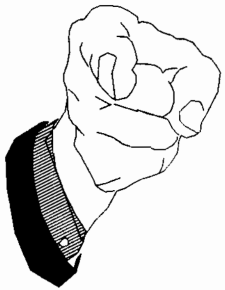
Dictionary
QUIZ
QUICK
QTP

HP/DEC/DG/IBM

- Incorporte Naming Conventions
- Test in PC Environment
- Port to HOST Environment
- Recompile Source for HOST
- Test in HOST Environment

Critique of DOS Version 5.11.F

- Sufficient Memory For Development



- Large Scale Applications
- Version 5.11.F has BUGs
  - Version 5.11.F2 EXT
    - Fixes to 5.11.F

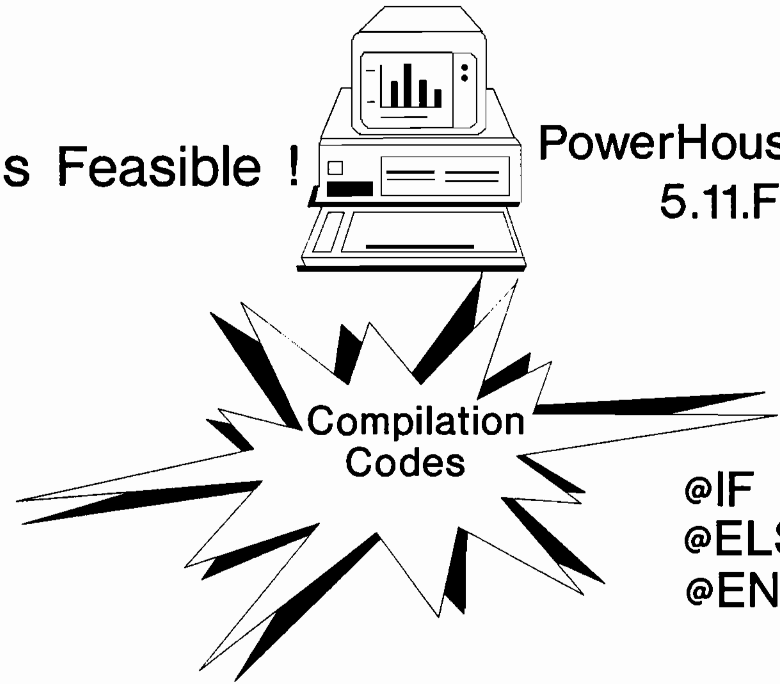
Cognos

Refer to SUPPORT Newsletter

Vol. 3 NO.1 January 1990

Developing Portable Applications

It is Feasible !



PowerHouse PC
5.11.F

@IF
@ELSEIF
@ENDIF

Developing a NewWave User Interface for MPE and HP-UX Applications

Mike Broadway

Pinewood Information Systems Division
Hewlett-Packard Ltd
Nine Mile Ride
Wokingham
Berkshire RG11 3LL
United Kingdom

Introduction

Hewlett-Packard is one of the leading suppliers in a new era of computing and information technology based on the Client-Server model, object orientation, and the graphical user interface of the NewWave workstation. While this NewWave Computing, or cooperative computing environment, promises a great many benefits it will take time to migrate existing applications to this new architecture. Many companies will be looking for ways to evolve towards NewWave Computing while continuing to maximize the benefit of their existing investments in hardware and software.

This paper introduces some of the tools available which can be used to assist in this migration process, extending the life of existing host / terminal applications while benefiting from something of the usability and distributed processing of NewWave Computing. While other tools will be mentioned in passing, the paper concentrates on the use of terminal emulators and the AdvanceLink family in particular. Examples are given of user interface customization, Dynamic Data Exchange for integrating host and workstation applications, Computer Based Training, and host control of the workstation.

NewWave Computing and the Client-Server Model

The conventional host / terminal model provides for shared access to a company's data at the cost of shared access to the computing resources of the single host computer, the more people using the system the slower it gets. The host computer is not only responsible for storing the data but also for its manipulation and the user interface, all at the same time.

In the NewWave Computing or Client-Server model, user workstations with their own computing resources are networked with shared machines which provide specialized services such as database management, printing, mail distribution, communications, general computation and so on. The NewWave data objects manipulated by a user at his or her workstation may reside elsewhere and be shared by many people, all transparently to the user.

Some of the servers may be conventional mini or mainframe machines, no longer responsible for supporting terminals and free to concentrate on database management, others might be single purpose hardware dedicated to a task such as 3-D graphical modeling. Servers may themselves be clients of other servers while workstations offer themselves as computation servers when they are idle.

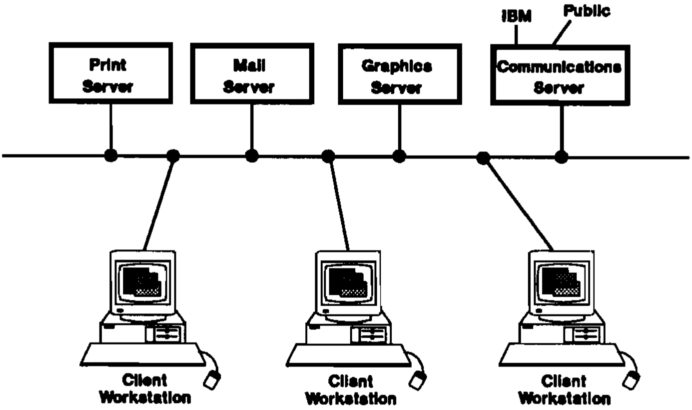


Figure 1 - The Client-Server model

A current example of a client-server application is NewWave Mail. The end-user reads, creates and stores his or her mail on their own NewWave workstation with its graphical user interface. This workstation is connected to a 'mail service' provided by HP Desk or OpenMail on a mini-computer. The user benefits from the ease of use of a graphical interface, integration of mail with word processing, spreadsheet and other workstation applications, and the greater responsiveness of the local CPU and disk resources. The mini-computer benefits from being able to support more users for the same computing power and central disc space.

One problem with the Client-Server model is the large investment that most companies have in existing centralized applications and databases, this investment can not simply be abandoned in the move to client-server computing. In some cases the centralized host / terminal model may remain in use for some time where a high degree of centralization is advantageous, Airline booking and banking systems for example. So how are these applications to be integrated with the NewWave Computing environment?

Some Migration Tools for NewWave Computing

If the host application is based entirely on HP 3000 VPLUS forms for its user interface then a quick and simple migration step is to use **VPLUS/Windows**. This product runs under Microsoft Windows, displaying and processing VPLUS forms in a window, connected to an HP 3000 through an HP OfficeShare LAN. This is not a terminal emulator but uses it's own protocol built on top of NetIPC. To the user it appears as just another Microsoft Windows application and allows information to be exchanged with other workstation applications via Cut & Paste.

The main limitations are that only host applications built entirely on VPLUS can be accessed and that a LAN connection is required. But with NewWave and other implementations planned, VPLUS/Windows offers a high degree of workstation independence in the near to medium term.

For those customers with more ambitious requirements and the resources to develop NewWave and Microsoft Windows applications themselves, **Cooperative Services** offers workstation access to HP 3000 IMAGE databases. The Cooperative Services package is based on object code libraries for PCs which effectively give PC applications access to the HP 3000's IMAGE intrinsics through an OfficeShare LAN connection. Using these object libraries NewWave applications can be custom designed to use the full range of window, menu and dialog box tools, and integrate local workstation databases with remote shared databases.

While in the longer term Cooperative Services and similar tools perhaps offer the best solution in terms of usability and integration they also involve the greatest investment and again require LAN connections. The new generation of **Terminal Emulation** products for Graphical User Interfaces (GUIs) may not offer quite the same level of transparent operation, but they do provide a fast and highly flexible approach to meeting immediate needs.

The New Generation of Terminal Emulators

The title 'terminal emulation' is really too limited in its scope to properly describe products like Hewlett-Packard's AdvanceLink, Tymlabs' Session, Future Soft's DynaComm, or Walker Richer & Quinn's Reflection. These programs now do very much more than simply emulate terminals. In a graphical user interface environment such as **NewWave** an emulator does not have to merely present a HP 700/94 terminal inside the frame of a window, it can improve on the original.

The windowing environment immediately allows for **multiple connections** to be supported and visible at the same time, using any combination of LAN and serial ports. Terminal system function key tree sets do not have to be slavishly copied where the same functionality can be offered through menus and dialog boxes. **Powerful script languages** like those of AdvanceLink and DynaComm allow for **user interface customization**, with menus specific to the host application in use and dialog boxes to question the user. **Host Control** facilities in AdvanceLink and Reflection extend the terminal's escape sequence capabilities to include most of the features of the script language. Support for Microsoft's **Dynamic Data Exchange** protocol permits integration with other workstation applications. Under NewWave, AdvanceLink allows scripts, and host applications using Host Control, to understand **Data Objects** and to invoke **Agent Tasks** and **Computer Based Training** (CBT).

The remainder of this paper consist of examples using most of these features as they are found in AdvanceLink. The individual examples themselves should not be taken too literally but they should provide a library of ideas applicable to real life applications.

Improving the User Interface of a Character Based Host Application

Somewhat artificially we will use HP Desk Manager as our main example host application but only because it is the host application that the author is most familiar with. We could equally well choose Materials Management (MM/3000), or any one of hundreds of third party and customer written host / terminal applications running under **MPE** or **HP-UX**. In fact HP Desk already supports two true workstation clients, AdvanceMail and NewWave Mail, which use HP Desk as their mail distribution server.

AdvanceLink under NewWave offers two types of data object: the **connection object** representing the terminal emulator together with the associated configuration information such as baud rate and communications port; and the **script object** representing command language scripts written to automate terminal use. Script objects may be permanently linked to connection objects in much the same way that a drawing object may be placed in a NewWave Write document. Such script objects can be configured to automatically execute when the connection object is opened, closed, or when a data object is dropped onto the open connection object window.

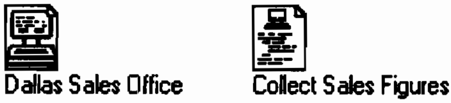


Figure 2 - AdvanceLink connection and script icons

A fairly simple AdvanceLink connection object can therefore be created which will automatically log on to a host computer and start up HP Desk whenever the connection object is opened through the execution of a script. Similarly the closing of the connection object window can invoke a second script to close down the host application and break the host connection. This most basic use of AdvanceLink is illustrated in figure 3.

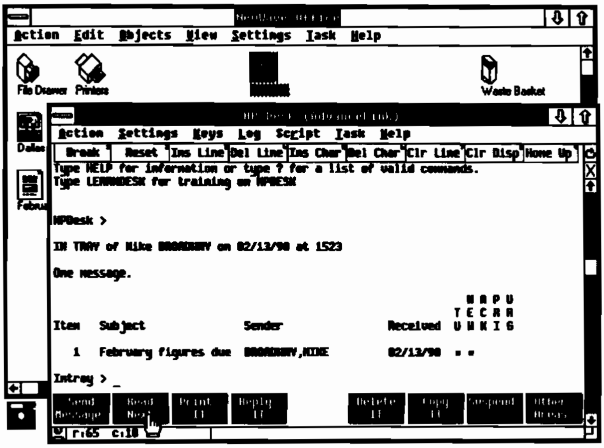


Figure 3 - Basic connection to HP Desk

Already the user has some usability improvements over a normal terminal: text can be cut and pasted between other workstation applications and HP Desk, the mouse can be used to click on the functions keys and controls like insert line, and the font and window sizes can be changed to show more information than the standard 24 by 80 character terminal screen.

The user is still however very much aware that he or she is using a terminal emulator, a workstation application whose menu and icon has no particular relationship to HP Desk. But we can change all this; the AdvanceLink script language includes commands for customizing the menu and window appearance. The following extract from the connection object's open script show how to tailor the menu for HP Desk.

```
; Turn off unwanted default menu items

remove menu "Settings"
remove menu "Log"
remove menu "Xfer"
remove menu "Script"

; Turn off the control bar to free screen space & remove confusing
; 'terminal' functions from user.
```

```
hide controlBar

; Add Tray / Area

add menu "Tra&y/Area"
add menu "Tra&y/Area" item "&In Tray" sendLine "intray"
add menu "Tra&y/Area" item "&Out Tray" sendLine "outtray"
add menu "Tra&y/Area" item "&Pending Tray" sendLine "pending"
add menu "Tra&y/Area" item "&Work Area" sendLine "work"
add menu "Tra&y/Area" item "&Filing Cabinet" sendLine "cabinet"
add menu "Tra&y/Area" item "&Calendar" sendLine "calendar"
add menu "Tra&y/Area" item "&Library" sendLine "library"
```

Changing the icon and creating an HP Desk system or 'tool' object may require a little more NewWave developer knowledge but is not especially difficult. We can define this by reusing the AdvanceLink program files but creating a separate NewWave installation control file. The installation of new objects is directed by a text file with the extension .IN$; installation is invoked either by pressing CTRL+ALT+X and selecting the Installation command from the menu or by copying the .IN$ file to \HPNWDATA\HPINSTAL.IN$ before starting NewWave and then loading NewWave.

Listed below is an example .IN$ file for such an HP Desk system object based on a edited copy of the standard NWADVLNK.IN$ file from the AdvanceLink installation discs. For more information on the meaning and format of installation files see the *HP NewWave Environment: Programmer Guide* and *Programmer Reference Manual* from the *HP NewWave Developer Kit*.

```
; Installation file for an AdvanceLink system object
;
; Replace 'HP Desk' in the following lines with the chosen name for
; your system object. These should be unique to this system object.

CLASS_NAME   "HP Desk"
ICON_NAME    "HP Desk"

; The TEXT_ID string will form part of the window title, enclosed in
; brackets to the right of the objects title.  It is a good idea to
; leave this as AdvanceLink so that any messages referring to
; AdvanceLink will make sense to the user.  If you change this text
; then the user may not immediately understand that the message
; refers to the terminal emulation window.

TEXT_ID      "AdvanceLink"

; Replace the 'HPDESK.ICO' text with the name of your own ICON file.

ICON_FILE    HPDESK.ICO       END_NAMES

; ** IMPORTANT **  The following GLOBAL_REF_NAME value must be unique
; for each system object on your workstation.  The default shown here
; is the same as the first value offered by the Bridge Builder
; application for user defined system or tool objects - if you need
; to change the value replace only the bottom four zero digits and
; not the 0x1fff.

GLOBAL_REF_NAME 0x1fff0000
```

```
; The following section will cause all files named TOOLDATA.* to be
; copied as part of the system object's data.  You should provide a
; default configuration data file named TOOLDATA.ACF on the
; installation disc.  This must be created using a normal
; AdvanceLink connection object and converting the configuration data
; to an MS-DOS file.
;
; You may optionally provide compiled script files with the
; extensions .OPN, .CLS, and .DRP to act as autoexecute scripts when
; the system object is opened, closed, or another data object is
; dropped on the open window respectively.

OBJECT_FILENAMES
    tooldata.*
END_NAMES

; The PROP_TITLE 'HP Desk' below is the title that user will be able
; to see when the system object is visible on the desk top and the
; name that the object's window will display.  The PROP_SYSTEM value
; indicates that the object is a system object and that its title can
; not be changed by the user.

OBJECT_PROPERTIES
        PROP_SYSTEM  BEGIN_VALUES 0x102 END_VALUES
        PROP_TITLE   "HP Desk"
END_PROPERTIES

; The following line will cause any object with the same
; GLOBAL_REF_NAME value to be removed from the workstation before
; this system object is installed.  This will be the normally
; required action to allow objects to be updated.
;
; If you are not sure which GLOBAL_REF_NAME values have already been
; used on your systems, you may comment out this line.  If an object
; already has the chosen reference name then the installation will
fail
; and the original object will be unharmed.

OVERWRITE_OLD_OBJECT

; The following MODULE_FILENAMES section can be omitted if the
; standard AdvanceLink objects have already been installed.  No
; matter how many system objects are created based on AdvanceLink
; there need only ever be one copy of these files on the workstation.
;
; If there are additional files, sub-scripts etc, which you need for
; the operation of your custom object they can also be named here and
; placed in a directory of your choosing.

MODULE_FILENAMES
  nwadvlnk.nwe
  *.fon          APP_DIRECTORY nwadvlnk
  hplink.*       APP_DIRECTORY nwadvlnk
  upload.c       APP_DIRECTORY nwadvlnk
```

```
END_NAMES

; ** DO NOT CHANGE ANYTHING BELOW THIS POINT **

EXECUTABLE_FILE  nwadvlnk.nwe

HAVE_METHODS
        CREATE_OMF
        OPEN
        WARM_START
        TERMINATE
        DIE_PLEASE
        OMF_INSERT
        QUERY_SHUTDOWN
END_METHODS

OTHER_CLASS_PROPERTIES
END_PROPERTIES
```

Note that at the time of writing the content of the AdvanceLink .IN$ MODULE_FILENAMES and HAVE_METHODS sections are not final and may differ slightly from those shown above.

Figure 4 illustrates the results of this work. Note the HP Desk icon at the top of the NewWave Office window, grayed because the object is open, also the custom menu and the extra screen space freed by the hiding of the control bar at the top of the AdvanceLink / HP Desk window.

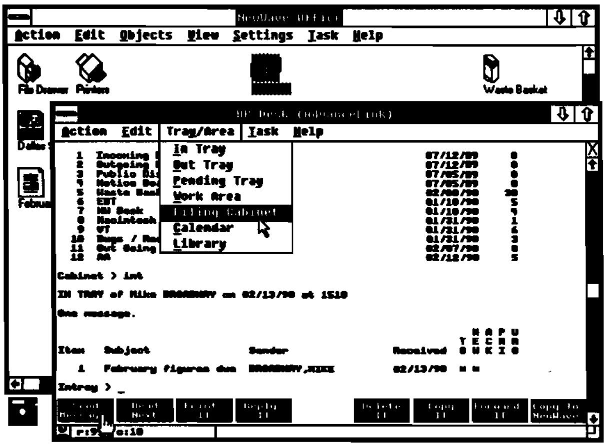


Figure 4 - Customized HP Desk window

Notice that with the removal of the Settings menu the end user is protected from accidental changes to the data communications settings such as the baud rate. For many users this will simplify the training required to use the host application and cut the internal support costs. A 'back door' is provided which allows the default AdvanceLink menu to be redisplayed for trouble shooting purposes.

HP Desk also offers some customization facilities that we can now take advantage of. Note that the F8 Other Areas key has been redefined as Copy to NewWave; with the 'other areas' listed in a menu this function key is no longer required to access a second set of function keys. The Copy to NewWave key can be linked to an HP Desk script written to open messages and transfer any NewWave objects contained in them down to the workstation. AdvanceLink offers escape sequences that can be used to invoke both NewWave object and MS-DOS file transfer, these sequences can be issued by the HP Desk script.

Drag and Drop

Any NewWave system object would be incomplete without the ability to deal with other NewWave objects. HP Desk is a mailing system, a user might reasonably expect to be able to send a data object from his or her workstation to other HP Desk users by dropping the object onto the HP Desk / AdvanceLink window and this can indeed be accomplished.

As mentioned above, an AdvanceLink connection object may have three autoexecute scripts associated with it; one each for opening, closing and handling dropped objects. When a data object is dropped on an AdvanceLink window will by default be uploaded to the host, if however a 'drag and drop' script is defined this will be invoked and allowed to decide what to do with the object. The following script would allow the user the choice, via a message dialog box, of sending the dropped object to another HP Desk user or simply storing it in the HP Desk work area.

```
; HP Desk 'Drag and Drop' script
;
; Extract the dropped object's title from its property list.  We are
; interested in the other properties such as its creation date.  The
; properties are obtained as a string with each property on a
; separate line in the string.  We use the 'chunking' facility to
; get just the title - the first line of the properties string.

name# = line 1 of objectProperties()

; Ask the user what to do with the object - send it or place it in
; the Work Area - by using a message box.  Note the use of the '&'
; character for continuation lines

messageBox into button# prompt name#+"^M^M"+ &
       "Do you wish to send this object to another^M"+ &
       "person or copy it to your Workarea?" &
               buttons "&Send", "&Workarea", "&Cancel"

; Now check the result of the messageBox and decide what to do.
; First, should we upload the object or did the user cancel?

if button# = 3                  ; if cancel button
  return                        ; .. end script
endif
send object "nwupload" delete   ; upload deleting any existing file

; Having uploaded the object decide whether the send it or place
; it in the workarea.

if button# = 1                  ; if send button
  sendline "send (nwupload)"    ; initiate the send process
  sendline name#                ; give the object's name as the
```

```
                                        ; message subject
    return                              ; stop the script leaving the user
                                        ; to complete the distribution list

  else                                  ; else it must be 'copy to workarea'
    set display off                     ; turn off screen changes - don't
                                        ; show user our workings
    sendline "suspend"                  ; preserve the user's current
  context
    sendline "work"                     ; switch to workarea
    sendline "copy from (nwupload)";    ; import the object to workarea
    sendline name$                      ; give object's name as item title
    sendline "resume"                   ; restore user's context
    set display on                      ; turn screen changes on again
  endif

  ; end of 'Drag and Drop' script
```

Obviously this script is a little naive since it does not check whether the transmission of the object was successful or not nor whether HP Desk in its current state would be capable of accepting the Suspend and Resume commands, but it serves as an illustration. In a real life implementation more care would have to be taken over such concerns. Figure 5 shows the dialog box display resulting from the NewWave Write object, 'February Sales', being dragged onto the HP Desk window.

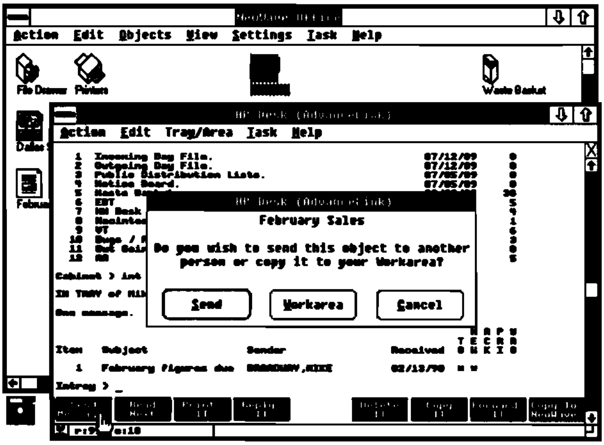


Figure 5 - Message box in response to 'dropped' object

Computer Based Training

Since the AdvanceLink connection object is fully integrated with the Agent and offers a class dependent suite of commands and functions in addition to its own script language, it would be a relatively straightforward exercise to link some form of Computer Based Training (CBT) to our hybrid HP Desk object using the standard NewWave CBT tools.

Such CBT courses could be invoked in the normal way through the Tutorial command of the NewWave Office Script menu. Alternatively AdvanceLink's script language includes facilities for launching agent tasks and so would allow CBT courses to be started from a custom menu command or at the request of the host application through a host control escape sequence.

The following AdvanceLink script sample shows how to invoke a CBT lesson located in the CBT global container, i.e. one normally accessable from the NewWave Office Task menu.

```
; Script to invoke CBT lesson stored in CBT Global Container
;
; First navigate through containers/folders to find the Agent Task
; object to be executed.  This is inside the CBT container named
; "HP Desk: Sending Mail" which is in turn located inside the
; CBT Global Container object.

set newWave location to CBTGlobalContainer
set newWave location to "HP Desk: Sending Mail"

; now select the Agent Task and execute it

select object "HP Desk: Sending Mail" type "Agent Task"
do agent task
```

Dynamic Data Exchange and Terminal Emulation

Both AdvanceLink and Future Soft's DynaComm offer powerful Dynamic Data Exchange (DDE) facilities. DDE is a standard protocol defined by Microsoft which allows for the exchange of information between two separate application programs running under Microsoft Windows and OS/2 Presentation Manager; in more general terms it is a form of Inter Process Communication (IPC).

In each DDE conversation there are two participants, a *client* and a *server*. Typically the client requests data from the server though it is possible to pass data in the opposite direction if desired. An example use of DDE would be to connect a word processing document to a spreadsheet, the document can then contain tables of figures extracted from the spreadsheet or even display a graph derived from these figures. When the spreadsheet data is changed the results will automatically appear in the word processing document.

DDE conversations are established by the client which specifies the name of the **application** it wants to talk to and the **topic** it wishes to talk about. Normally the application name would correspond with the product name of the server, 'Excel' for example. Topic names are defined by the server application and can relate to anything that makes sense for the server; Excel topics correspond to spreadsheet names, DynaComm topics are the names of scripts which should be run. AdvanceLink uses the name of the the configuration file or under NewWave, the connection object name.

Once the conversation is established the client can name **items** available under the given **topic** and either request the data associated with that item or 'poke' data into that item. Note that the client is always the active participant in the conversation. If two way communication is required then two separate DDE conversations must be started with each application acting as both client for one and server for the other.

We will now examine three different uses of DDE in conjunction with terminal emulation. The first two scenarios are described in general terms while the third will introduce some of the script commands provided by AdvanceLink to support DDE.

Hiding Terminal Emulation Behind A Preferred Application

Training and support can be an expensive process; hence the fewer applications a user has to be familiar with the better. There are many useful information sources available through terminal emulation and dial-up connections but each of the services introduces another application requiring training; a well written command script and DDE can allow users to access such information through the interface of their existing tools.

For example, an investments manager may be trained in the use of Microsoft Excel but not in AdvanceLink terminal emulation or the dial up stock price information service. Excel macros can be written for this user which add new menu items like 'Update Prices', when this is selected Excel can launch AdvanceLink together with a script file. The script will dial up and log in to the stock price service and then establish a DDE conversation with Excel. Excel can then obtain the required prices without the end user having to interact with the terminal emulator at all, the only interface the user deals with is Excel's.

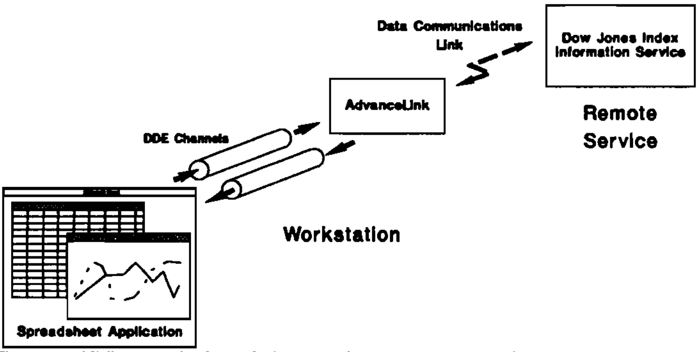


Figure 6 - Hiding terminal emulation and the host application behind Microsoft Excel

Combining Local and Remote Applications

Newer applications will tend to be implemented to the Client-Server model, indeed their very nature may require it, but large amounts of information will remain on host / terminal systems for some time to come. Terminal emulation and DDE offer a means to integrate the old with the new.

A good illustration of the value of such integration is provided by Hewlett-Packard's AIMS (Advanced Image Management System) product. AIMS allows databases of mixed text and image information to be created and stored on an HP-UX server with multiple Microsoft Windows workstations on a LAN acting as display clients, these allow both the text and images to be entered, selected and viewed. A powerful fourth generation language aids the creation of custom applications, this language includes support for DDE.

Where a company has a large existing database application on a host / terminal system it may be too expensive or impractical to equip every employee with the required graphical workstation and LAN connections to completely replace it with an AIMS system. Instead the existing system may be integrated with AIMS to provide the additional pictorial information only to those within the organization who need it.

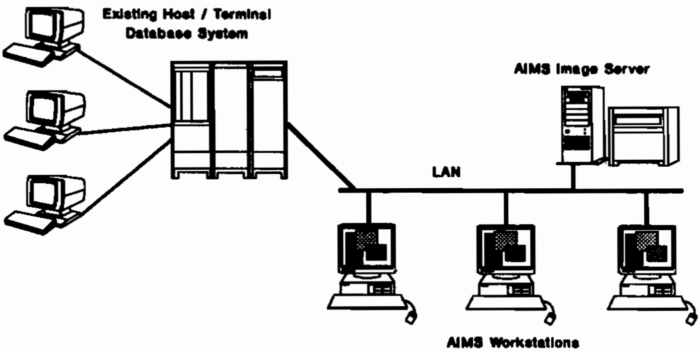


Figure 7 - Integration of Client-Server and Host / Terminal systems

An insurance company might continue to use its existing computer systems to track policies and provide information to the whole company, while storing images of all the written correspondence associated with these policies in an AIMS database. Users of the AIMS system can browse through policy records on the host data base using a terminal emulator, when they find a particular policy they can call up the stored images of the original application, letters, payment cheques and so on relating to that policy. Other examples include parts catalogs with images of the items and personnel records with photographs, hand written appraisals and such.

DDE is used to link the 4GL generated AIMS application to the terminal emulator. To call up the images associated with the currently displayed record in the terminal display the user presses a button in the image application window or a terminal window function key. In either case DDE is used to pass some of the terminal's data fields to the AIMS application, the policy number and surname of the holder perhaps, these are then used as keys into the AIMS database to retrieve the document images.

Developing Client-Server Applications on DDE and Terminal Emulation

Terminals emulators supporting DDE can be used as the basis for the development of true Client-Server applications, providing a transparent data communications link to the server. The workstation client application can be written fairly quickly in one of several fourth generation languages, such as that used in the AIMS system or the Whitewater Group's Actor, and use DDE to link to the emulator. The client application does not require any knowledge of the data communications, serial ports and the like, these are handled by command scripts written for the emulator.

Returning to our HP Desk example once again, it would be possible to write a collection of simple Microsoft Windows applications which were able to represent the various In Tray, Out Tray, Library, and other areas of HP Desk as list boxes with one line for each message or item; double clicking on an entry in one of these list boxes would display the appropriate item text in a separate window.

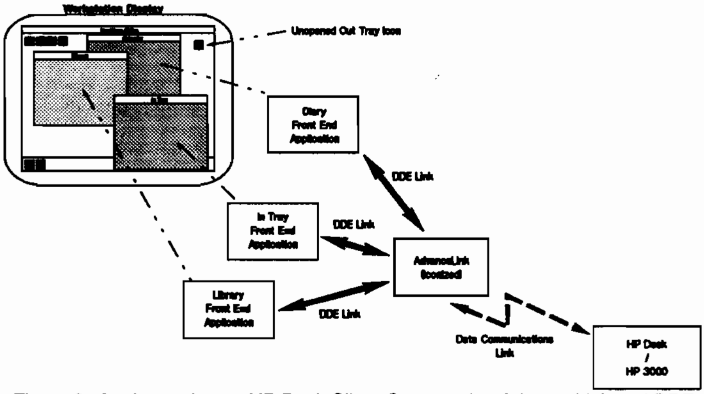


Figure 8 - Implementing an HP Desk Client-Server using AdvanceLink and DDE

Any time one of these small applications is loaded by the user it will first check to see if AdvanceLink is already running and offering the HP Desk connection. If not the mini-application will launch AdvanceLink with a command script to establish the host connection, log on to HP Desk, and set up the DDE service routines for listing and reading messages. The following script sample is perhaps a little careless about log on errors but might serve as a prototype:

```
; Iconize AdvanceLink, Log on to HP 3000 over LAN and start HP Desk

minimize
set hostPrompt ":^Q"
connect "A950.PWD.HP"
sendline "hello michael/wombat.broadway/mouse"
sendline "hpdesk 0 broadway"

; Change the DDE application name to reflect our specific purpose

set ddeApplication to "HP Desk Link"

; Define DDE service handlers for accessing mail messages

on DDE poke "TRAY" do script "desktray"
on DDE request "READ MESSAGE" do script "deskread"
on DDE request "MESSAGE COUNT" do script "deskcnt"
on DDE poke "MESSAGE NUMBER" do script "deskpos"
on DDE poke "DELETE MESSAGE" do script "deskdel"
on DDE poke "SEND MESSAGE" do script "desksend"

; Enable the DDE handlers

set ddeServer on
```

The `set ddeApplication to "HP Desk Link"` command changes the application name that the terminal emulator will respond to when a client first seeks to initiate a DDE conversation. By default this would be 'AdvanceLink' but this would not allow the client to differentiate between multiple copies of the emulator running at the same time, so one of the first thing our script does is change the name to something specific to the service being offered by this instance of AdvanceLink.

The various `on DDE` commands define scripts which will be executed when certain DDE messages arrive from the client. DDE POKE messages allow the client to pass data to the server, in the case of a spreadsheet server the item would typically correspond to a spreadsheet cell; DDE REQUEST messages ask the server to return a specified item of data to the client, again in the case of a spreadsheet the item would probably be a cell or range of cells.

In our example the client can switch to the In Tray of HP Desk by poking the text data 'In Tray' to the 'TRAY' item, when this message is received AdvanceLink will execute the script named DESKTRAY which can examine the poked data and respond accordingly. The client can ask for a count of messages in the tray by sending a DDE REQUEST message to the 'MESSAGE COUNT' item, this will cause the DESKCNT script to be run which will find the answer from HP Desk and return it to the client.

Note that items do not have to directly correspond to some piece of information stored in the terminal emulator application but can refer to the current mode or state of the host application; because AdvanceLink handles DDE messages by executing scripts we can define 'items' to suit almost any purpose.

The scripts for handling the poking of data to the MESSAGE NUMBER item and the requesting of data from the READ MESSAGE item are listed below. These two handlers allow the client to specify the message to be read and then obtain the message text.

```
; DESKPOS.ASC
;
; DDE POKE handler for 'MESSAGE NUMBER' item for 'HP Desk Link'
;
; The client pokes data into this item to change the current message
; number. If the supplied message number is larger than of the last
; message in the tray then the DDE POKE is rejected.  N.B. the
; msgCount variable is set to the size of the tray by the DESKTRAY
; script when the tray is first selected by the client

if num(ddeData()) <= msgCount#
    msgNum# = num(ddeData())
    DDE respond accept
else
    DDE respond reject
endif
```

The `ddeData()` function is used by the script to obtain the text sent by the client application to be 'poked' into the MESSAGE NUMBER item. The `DDE respond` allows the script to inform the client whether or not the DDE message was accepted and processed correctly.

```
; DESKREAD.ASC
;
; DDE REQUEST handler for 'READ MESSAGE' item for 'HP Desk Link'
;
; The client should already have set the msgNum# message number to be
; read via a DDE POKE to the 'MESSAGE NUMBER' item
;
; The variable trayPrompt# will contain the appropriate prompt string
; for the current tray, e.g. "In Tray>^Q" and is set by the DESKTRAY
```

```
; script in response to DDE POKEs of tray names to the 'TRAY' item.

sendline "read "+str(msgNum#) expect trayPrompt# into msgText#
DDE respond msgText#
```

The DESKREAD script illustrates the use of one of the more powerful commands in AdvanceLink's script language. If the `msgNum#` variable contains the number 3 then the `sendline` command will send the string "read 3" together with a carriage return to the host application, HP Desk. The command then `expects` or waits for the string contained in the `trayPrompt#` variable; in our example this would contain the prompt displayed by HP Desk when it has completed the display of a message. Until this prompt is seen all the text sent by the host is logged `into` the `msgText#` variable. AdvanceLink variables can store up to 32K bytes of text which should be more than enough for the average HP Desk message.

In this script the `DDE respond` command takes the `msgText#` variable as a parameter and returns its contents to the client.

With NewWave Mail already providing a true client-server HP Desk based mailing solution in a much more elegant fashion, with better performance and less drain on host CPU resources, it is unlikely that anyone would use DDE and a terminal emulator to do this job. In other cases these facilities may provide a short or medium term solution not otherwise available.

Host Control

Host Control is the title given to the AdvanceLink facility which allows host programs to pass script commands to the terminal emulator for execution, the command text is simply encapsulated in an escape sequence and sent to the terminal. Almost all the functionality of the script language is thus available to extend the escape sequences of the emulated terminal. Since the script language includes commands for manipulating MS-DOS files, NewWave objects, and launching NewWave Agent Tasks, the host application can virtually take control of the workstation. Even DDE can be used by the host application.

A combination of the NewWave Agent and Host Control could be used to automatically update workstations and collect data from them each night. AdvanceLink could be started by an Agent Calendar task outside normal working hours and log on a central administration host computer. Once the connection has been opened the central computer can take control of the workstation using host control and perform a variety of housekeeping tasks. The advantage of using Host Control is that the set of tasks to be performed can be centrally administered and changed from day to day as required.

The host application can invoke back up procedures, either through a separate Workstation program or by selecting specific important files and NewWave objects itself and transferring them to the host. Documents such as the latest price list or sales data can be distributed, appearing in the top level NewWave Office window when the user returns the next morning.

With DDE, host and workstation databases can be integrated. We have already seen how DDE can be used to give workstation applications access to host data, but with Host Control in combination with DDE the host application can reverse this process. The latest figures held in a local stock controller's workstation spreadsheet can be automatically collected while the most up to date pricing information is fed back into the same file.

Computer Museum

The Not So Distant Future

This paper has attempted to show something of the potential of a new generation of terminal emulators built on graphical user interface environments. Whether it is simply to cut and paste text between host and workstation applications or to completely hide the host computer and terminal interface behind DDE, these new emulators can be used to increase the usability and value of existing host applications and provide a bridge to the client-server architectures of the 1990s. All of the facilities described are available today, but what of the future?

The death of the terminal has been predicted almost as frequently as the end of the world, and with as much accuracy. Some have even begun to predict the demise of the terminal emulator as well, but this is premature.

Terminal emulators will evolve over the next couple of years to become their own standard, instead of simply emulating terminal hardware they will define more and more of their own unique functionality. Apple's MacWorkStation product already illustrates something of this, offering escape sequence control of menus, dialog boxes and child windows. It will soon be possible to develop user interfaces for host applications which are indistinguishable from the workstation applications they run beside.

With the proliferation of workstation operating systems, Microsoft Windows, OS/2 Presentation Manager, X-Windows, Apple Macintosh and more, terminal emulators have the advantage of being platform independent. As they evolve into user interface servers rather than just emulating terminal hardware they will assist in the integration of the user community and reduce development costs.

Business Session is a registered trademark of Tymlabs Corporation.
Microsoft, Windows, MS-DOS and Excel are registered trademarks of Microsoft Corporation
OS/2 is a trademark of Microsoft Corporation and International Business Machines Corporation
DynaComm and Future Soft Engineering, Inc. are registered trademarks of Future Soft Engineering, Inc.
Reflection is a registered trademark of Walker Richer & Quinn, Inc.
Apple and Macintosh are registered trademarks of Apple Computer, Inc.
MacWorkStation is a trademark of Apple Computer, Inc.

TITLE: Distributed Application Processing

AUTHOR: Charles Warzecha

Gateway Systems Corp.

2400 Science Parkway

Okemos, MI 48864

517-349-7740

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 4049

File Serving with the HP3000

Bud Booth
AA Software Development
429 S. Huntington Ave.
Boston, Ma 02130
(617)522-6531

This paper describes techniques for extracting data from an HP3000 and transferring the selected data to a remote PC.

What is a file server?

A file server is best defined by the environment in which it will be used. In the most general sense, the file server holds information in files. The files are updated by a user from an input device. Information is obtained from the files by requesting the information from an external device, and is forwarded to the same device or to an auxiliary device.

In the environment where a user is linked to an HP3000 by a terminal, data is sent to the HP3000 or retrieved by commands issued from the remote terminal. Appropriate commands select the data and direct the data to the terminal, to a disc file, or to a printer. Files are shared by appropriate permissions and the integrity of the data is protected by locking a data item, record, or file. Software resident on the HP3000 provides instructions to the HP3000, which processes the request.

In an environment in which the user is operating an independent computer processor, such as in a network of computers, the remote device contains appropriate software to allow the remote computer to transmit or request data from a remote file. It may also contain it's own software for further processing data.

Software is available to allow a remote computer to send requests for individual data items or records from a "file server". One type of software is called SQL (System Query Language). The remote computer sends a request to the file host. The host processor satisfies the request using it's own software, and accepts data or returns data to the remote computer, which utilizes it's own compatible software to maintain or operate on the data.

This paper describes another scheme for selecting data from the file server. This does not require specialized software on the host (in this case the HP3000). It utilizes existing host files and programs, and does not require "re-filing" or reorganizing data with any specialized system or software.

Environment

Southern California Gas Company is a utility which provides gas to homes and industrial users in California communities. The gas is purchased from suppliers in California, as well as other suppliers throughout the U.S. The gas is transported through pipelines originating in many places in the U.S. and Canada, and is delivered to the company at a number of locations in California.

Each delivery point has measurement devices which record parameters such as volume, pressure, and time. Additional measurements are made at storage locations, and at various junctions throughout California. Measurements are made at one minute intervals at each of the 850 measuring locations in the gas system. The measurements are transmitted to an HP 1000.

The HP 1000 is programed to display each section of the pipeline with the current volume and pressure at each point. Operators monitor gas flow and control gas flow in the system based on these on-line displays.

Once each hour, the HP 1000 calculates the average measurement for each location, and transmits this data to an HP 3000.

Measurement data is stored in a circular file, so that once each hour, the new data is added to the file, and the oldest records in the file are effectively deleted. The file accommodates 2 years of data. Capacity has been provided for up to 1200 measurement points per record. The file contains 1 record for each of 24 hours per day x 720 days = 17,280 records with 20,738,000 measurements.

A management decision was made to incorporate this data, along with all customer related and operational data, on an IBM mainframe. Ultimately all data would reside on the mainframe and would be accessible to users throughout the company.

The task at hand was to transfer selected data to the mainframe accurately and effectively.

System and Software Requirements Specifications

A. Functional Requirements

1. From a remote computer, select data based on:

   a. Specified measurement point. From 1 to 1200 different measurement points may be selected.

   b. Specified dates. From any date within the last 2 years to any date within the last 2 years. Total number of days will be 1-720 days of data.

2. Data may be requested from the remote PC using a manual request mode. Data may also be requested in an automatic mode at pre-determined times.

B. Performance Requirements

1. Data transfer must be transparent to users on the HP3000 with minimum delay in system access. Total delay to be less than 10 minutes per day.

2. Degradation of system performance must be minimized during transfer operation.

C. Design requirements and constraints

1. The current programs on the HP3000 have been operational and trouble free for 2 years. The data file structure is sound and is designed for compatibility with the HP1000 programs which update the files. No changes are to be made to existing programs or files.

2. All HP3000 programs and HP1000 programs are written in FORTRAN. The programming staff is familiar with FORTRAN. New programs must be in FORTRAN for compatibility and maintenance.

3. Data will be "down-loaded" from the HP3000 to an IBM PC/AT with 1MB of memory and 40 MB hard disc drive.

4. Data will be transferred to an IBM mainframe over an IBM Token Ring network, using APPC (Advanced Program-to-Program Communication) software.

D. Validation tests

1. A permanent test program or procedure must be written and be available to validate all data transferred from the HP3000 to the IBM PC and from the IBM PC to the IBM mainframe.

2. "Stress" tests are designed to simulate or test the program when usual or unusual events occur. These include such events as:

   a. HP3000 failure
   b. Backup in process
   c. Files already in use
   d. Communication failure

System Design

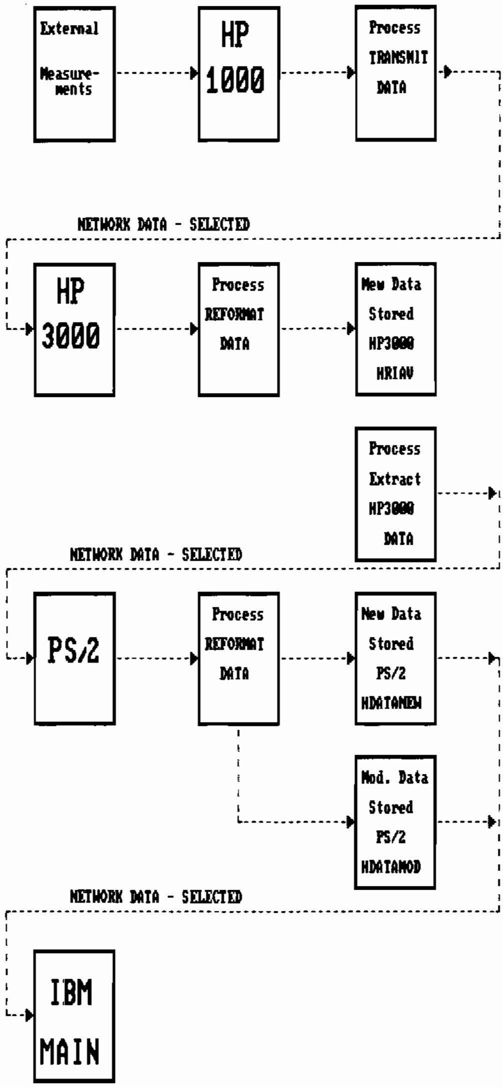
The total system can be divided into 3 sub-systems. The first sub-system is concerned with the selection and extraction of the data from the HP3000 data files and preparing the data for transmission to the IBM PC.

The second sub-system operates from the IBM PC and provides information to the HP3000 to identify the data to be selected from the HP3000, control the HP3000, transfer data to the PC and allow the PC to process data requests and to receive transmitted data.

The final component is a sub-system to transmit the data from the PC to the IBM mainframe.

Figure 1 shows the flow of data through the system.

Figure 1 - Flow of data through the System



Interface table

Rather than change existing software which manages and processes data and data files, it is usually more desirable to extract data from a program external to the software.

In order to accomplish this, it is first necessary to identify the "locator" for the data. A locator is a field or set of fields which uniquely identify the data item or record of interest.

File systems and database systems often provide the capability to use keys to locate records. Keys may be unique or they may identify a set of records. In KSAM and IMAGE, for example, a key may point to a "chain" of records which contain the key. This is useful for quickly getting "close", and then doing a search through the chain for a second identifier. The "locator" would consist of the key + the second identifier.

Example:

Check number - This can relate vendor payments to bank accounts, invoices, or vendor account activity. In some cases the check number could be the locator. In other situations, the check number + the invoice number might be the desired locator.

An interface table is an external file designed to contain the locator necessary to extract unique records or fields from an existing data file.

Extraction Program

By studying the structure of the file containing the data to be extracted, it is possible to identify the keys and "locators" available for accessing data records and fields. The current programs for maintaining the data should be studied as well.

The portions of the existing program related to accessing of data can be copied. If they are not already subroutines, then new subroutines can be created using the existing code.

The subroutines can then be modified to use the locator to access those records which are needed. The extraction subroutine:

1. Refers to the interface table for the desired locator.

2. Accesses the record or records of interest using existing keys.

3. Extracts records or data fields based on additional criteria defined by the locator.

4. Writes the extracted data to a new file.

A new program for creating and updating the interface table may be written as a stand-alone program. The subroutines for accessing data are included in the extraction program.

Application

For this project, it was necessary to locate a record for each measurement point for each hour of a specified date.

The master data file contains one record for each hour of each day. Each record contains the measurement point number, followed by several measurements. The file is a circular file.

The address of each record in the file changes with each new record entry. In order to minimize access time, an algorithm was written to calculate the address. Given the date and hour of the first record, and knowing that 24 records are added each day, the address of any record can be calculated. The algorithm also makes the adjustment necessary on dates associated with daylight savings time (Moving the clock forward or back 1 hour results in 23 hours or 25 hours on these dates).

Using the interface table locators

In this application, the "master" interface table consists of:

1. An identification number for each measurement point. This number identifies a data field in each data record.

2. A description of the measurement point. This is used only for additional identification on input screens or output reports.

3. A letter indicating whether the measurement is a pressure (P) or volume (V) measurement.

4. A field which is used to indicate whether to extract data for this measurement point or to ignore it. The purpose of this field is to allow an infinite number of selections to be made. The master interface table is copied to a "working" table.

The user uses a selection table maintenance program screen to indicate which points are to be extracted by entering Y or N in this field.

The extraction process

The extraction program executes in either a manual or automatic mode. In the automatic mode, these parameters are read from an ascii file. The file is always saved with the name "INCOM". If the file exists, parameters are read from the file. If it does not exist, then the extraction parameters are entered interactively on a screen.

The parameters required are:

1. The name of the interface file to use. As described above, the interface files may created at any time and contain the identification of the measurement points of interest.

2. The date (month,day,year) on which to start the selection process.

3. The number of days of data to select. Note that data for each hour of each selected day is extracted.

4. The name of the file for storing the extracted data.

When the extraction program is executed, the program:

1. Reads the interface file and stores the identification number of those points marked "Y" into a 1 by 1200 array to be used as the selection list.

2. Locks the master file and locates the date to start reading.

3. Selects the first record and stores the measurements in a one dimension array. The 1st position represents measurement point 1, the 2nd is point 2, etc.

4. Reads the 1st item on the selection list.

5. Writes a record with the date, hour, identification number, and the value of the measurement to the output file.

6. Repeats step 4 and 5 until all items are selected.

7. Repeats steps 3 to 6 until all hours for all dates have been read.

This process requires only 1 read operation on the master data file for each hourly record and minimizes the total time required to lock the master file. Because a circular file is volatile, it was necessary to lock the file. In more common applications, it probably will be adequate to lock only 1 record, which would result in less interference with production operations.

In operation, data for 1 day is selected and written to a file in an average of 1 minute.

Extraction process summary

The methodology used for this project can be applied to a wide variety of projects. It is independent of the language used for existing programs, and is independent of the type of file currently storing the data of interest. Moreover, it does not require changes to existing programs or data files.

The use of existing data access code reduces the opportunity for coding errors. If the file structure is changed, updating of the extraction program will be much less painful.

Remote operation

The second phase of the project is to enable a remote PC to perform the data extraction and transfer the data from the HP3000 to the PC.

There are a number of software programs that allow remote access from PC's to other PC's or mainframes. HP Link was selected as a company standard and is to be used for this project to interface from the PC to the HP3000. APPC (Advanced Program-to-Program Communication) is to be used to link to the IBM mainframe.

Tasks

Two different types of tasks must be performed. The first task is to allow a user to extract performance data from the HP3000 for specific measurement points over a specified period of time. This requirement arises at irregular times when special studies or analyses are required. Examples include:

Analysis of a section of the network over periods of days, weeks, months, or years. This requires selection of specific measurement points over a given period.

Analysis and comparisons of specified measurement points to study seasonal effects, or effects of meter types.

Special studies are made as follows:

1. Using a screen on the PC, the user indicates that he (or she) wants to create a new interface table.

2. The master interface table is displayed and the user scrolls through the table, marking the desired points with a Y. The new table is saved in a file.

3. A second screen presents a list of the required parameters. The user indicates the name of the file which will contain the parameters, and enters the name of the interface file, the date to start, and the number of days to extract.

4. The user exits the "request" program and executes a batch file which signs on the HP3000, extracts the desired data, transmits the data to the PC, and signs off. Details of the process are more fully explained below.

Automatic Mode

Data must be transmitted to the IBM mainframe each day. The time selected for the transfer was 8:00AM. This is the start of the day shift, and interference with normal computer operations would be minimal.

The IBM mainframe sends a message via an APPC program to the IBM PC. This initializes a batch file which "wakes up" the PC. The PC signs on to the HP3000, obtains the standard set of data, signs off the HP3000 and transmits the file to the IBM mainframe. Finally, the PC detaches from the IBM mainframe and goes to "sleep" until the next day.

PC to HP3000 Interaction

HP LINK provides an excellent vehicle for this interaction. The program has excellent "script" capabilities allowing the programmer to customize instructions using the HP LINK programming language.

The HP LINK program allows the PC to send commands to the HP3000, and to read the HP3000 responses directly from the screen, just as the user would read them. The programmer can then control the actions of the PC as well as the HP3000 based on interpretation of these messages.

HP LINK also allows the option to "chain" to a new procedure, exit, or return to a calling procedure. Conditional statements thus allow a great deal of flexibility and control over a process.

One more significant feature of HP LINK lies in the ability to "build" text strings within the program. These strings may be used internally for conditional checking or sent to the HP3000 as a command.

Process - Sign on to HP3000, extract data to a file, and return file to PC.

Step 1 - Sign on to HP3000. This procedure signs on to the HP3000. The procedure reads the HP3000 responses. It tests to see if someone is already logged on, as well as testing for "Can't initiate new sessions". If unable to log on, the procedure provides a message to the PC, waits an appropriate amount of time, and tries again. The procedure repeats this until log-on is successful.

Step 2 - Send the command file to the HP3000. This sends the file containing commands and parameters which will be used by the Extraction program resident on the HP3000.

Step 3 - The standard Interface Table resides on the HP3000. If a special table is to be used, the PC copies the special table to the HP3000.

Step 4 - Send a command to execute the Extraction program. This executes the Extraction program, which in turn reads the command file, selects the appropriate data, and saves the data in the file specified by the PC.

While the processing is going on, the PC waits for a pre- defined message "Selected Data from HP3000". This message is written to the screen by the Extraction program after the program successfully executes. If the Extraction program encounters problems which it cannot handle, it send the message "Cannot complete" to the screen. The PC ignores any other messages.

Any problems encountered by the Extraction program on the HP3000 are handled there. For example, if the file is already in use, the Extraction program delays, and tries again until it gains access, and successfully completes it's work.

Step 5 - When the PC detects the completion message, it copies the file with the extracted data to a local directory. The PC also copies the Interface file used for the selection process. This provides a means to verify the contents of the extracted data at any time.

Step 6 - The PC logs off of the HP3000 and returns control to the batch file on the PC. The batch file returns control to the APPC commands file which transmits the data to the IBM mainframe. The APPC commands file then places the PC in a "wait" mode, so that the PC is ready to receive commands from the IBM mainframe.

The PC may be used for other purposes by exiting from the APPC command file. At any time, the APPC command file can be restarted, so that the PC will be ready for IBM mainframe commands.

Summary -

This paper has provided a "real life" example, where the HP3000 is used as a repository for a large amount of data. The HP3000 acts as a "collector" for the data, which is used for a number of purposes in the HP3000 environment.

Further use of selected data is enabled by providing a simple means for extracting the data from the HP3000. The integrity of the data and the existing programs has been preserved by "attacking" the data with external programs, and preserving the structure of the file containing the original data.

Fortran was used for this project because this was a "Fortran" shop. However, the techniques used can be used effectively with other languages.

Microsoft Fortran 4.1 was used to develop some of the sub- routines which were transported to the HP3000. Fortran IV on the HP3000 is very compatible with the Microsoft Fortran. Conversion of most syntax differences (primarily $ {control} statements) were quickly changed using "replace all" features of the IBM Personal Editor program.

I personally found that the Microsoft Fortran provides an excellent programming environment. Their programming editor allows you to compile within the editor. When a syntax error occurs, an error message appears at the bottom of the appropriate page, and the cursor is positioned at the exact location of the error. Microsoft Codeview allows statement by statement execution of the program. Values can be changed, and variables can be "traced" and their values can be displayed as they change during execution of the program.

As previously described, HP LINK provides many features and capabilities to allow the programmer great flexibility in designing the intra-computer procedures.

Hopefully, the readers can use the techniques described here as a guide to creating solutions to assist them in using the HP3000 to serve their own PC's.

A number of trademarks and registered trademarks appear in this paper. The companies listed below own the marks preceding their names.
HP LINK, HP3000: Hewlett Packard; Microsoft Fortran, Microsoft Codeview: Microsoft Corporation; APPC, IBM Corporation.

TITLE: Implementaing NewWave - A Case Study

AUTHOR: Don Lewis

The Apex Group

7151 Columbia Gateway Dr.

Columbia, MD 21046

301-290-1606

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 4066

The HP3386 and HP3286
Integrating the HP3000 with the PC
The Next Generation of Computer Systems

by

Leroy Friesenhahn
Blanket Resources, Inc.
12337 Jones Road
Suite 408
Houston, Texas USA 77070
(713) 469-0869

In August 1981, IBM introduced its personal computer. Its introduction began a new era of information processing. The PC soon became the de facto standard for the business work station. The introduction of the 286 PC and the 386 PC has provided businesses with a more cost effective way of processing information. Studies have shown that it is 30 to 100 times more expensive to process information on a mini computer than on a PC. This is not to say that all processing should be moved from the mini computer to the PC, but that the task should be delegated to the mini computer, the PC, or both, based on the task requirements. Access to large amounts of information by many individuals is best performed by the HP3000, while "what if" analysis of selected information is more efficiently handled by the PC.

The PC, with its user friendly software and cost effective processing powers, has changed from a business luxury to a necessity. The tremendous influx of PCs into the workplace and the high cost of centralized computer systems has resulted in many companies maintaining multiple computer systems. A centralized computer system, the HP3000, is used to perform traditional data processing functions, while PCs or PC networks are used to perform individual or department tasks. The power of the 286 or 386 computer is often under utilized while the resources of our HP3000 are stretched to the limit. The purpose of this paper is to explore the traditional roles of the PC and the HP3000 and to discuss additional methods of shifting selected information processing from the HP3000 to the PC.

Various tasks require different resources in different quantities. To obtain maximum performance it is important to develop different strategies for solving different types of tasks. One process may be CPU intensive, while another is heavily I/O bounded. Another process may require a burst of processing, such as painting a user's screen, and then wait for input. To each of these situations, a different method is needed to effectively use the HP3000 and the PC together in completing the task. The processes used between the HP3000 and the PC to handle various tasks are divided into five different methods. Each method has its advantages and disadvantages, therefore, the desired task must be analyzed before selecting a method to apply. These methods are defined as follows:

1. Stand Alone Processing	- Moving data from the HP3000 to the PC for processing
2. Off Load Processing	- Processing performed by the PC but managed by the HP3000
3. Share Processing	- The PC provides information for a job on the HP3000
4. Cooperative Processing	- The PC performs a job using the HP3000 as a source for information
5. Parallel Processing	- A job is divided into tasks with the HP3000 and PCs performing these tasks at the same time

The following discussion defines each of these processes in detail and develops examples of practical applications.

Stand Alone Processing is defined as the method by which a PC uses data independently accumulated on the HP3000. Data on the HP3000 resides within MPE files, KSAM files or IMAGE. Although the procedures for handling the three types of data structures are slightly different, the following discussion and example will focus only on the use of IMAGE. The process can be divided into three steps.

Step one consists of collecting the desired information on the HP3000. Query can be used to perform this task, but commercially available programs, such as Data Express, provide an easier way of collecting this information. Figure 1 shows the various sources used by Data Express to define the HP3000 storage structure. In this example, Data Express will use the HP IMAGE root file definition. Figure 2 requests the location of the data base, group and account, and password needed. The desired data sets are selected as shown in Figure 3. The desired fields within the selected data sets are chosen as shown in Figure 4. Figure 5 shows the selection criteria used in determining the information to be contained in the output. The fields included in the output, sort order, summarization, and selection option are specified. The process continues with the selection of the desired file formats needed by the user. Figure 6 shows the different file formats available.

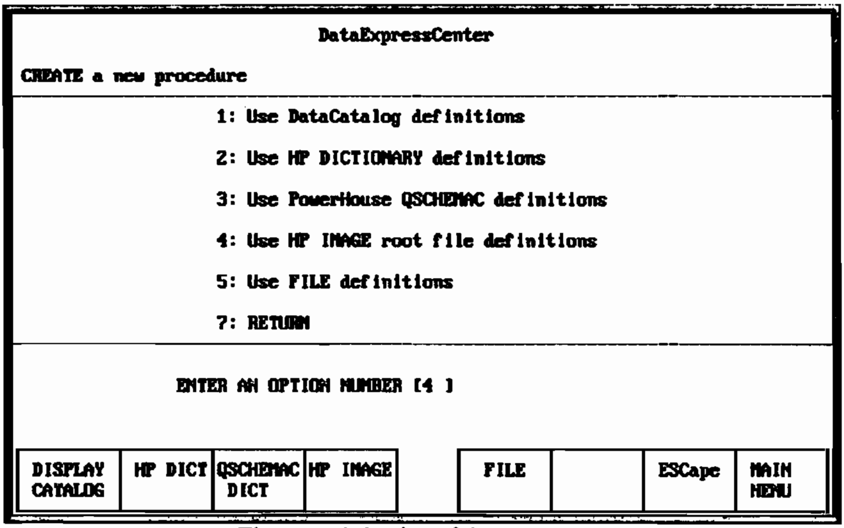


Figure 1 - Selection of data structures

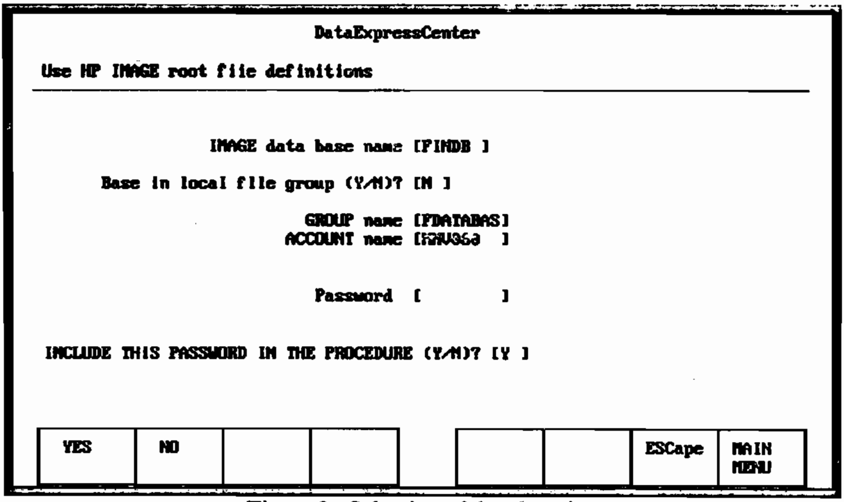


Figure 2 - Selection of data location

```
                              DataExpressCenter

Data sets in data base FINDB.FDATABAS.MMV060

  1: PTMAS                 14: OPTMAS               27: CMFIL
  2: PROMAS                15: CURPRMAS             28: SMFIL
  3: ALOMAS                16: CIAMAS               29: GLDFIL
  4: SALMAS                17: PRTMAS               30: CRFIL
  5: SHPMAS                18: ALDFIL               31: CIAFIL
  6: BILMAS                19: CUSFIL               32: CHAFIL
  7: SCTMAS                20: SOEFIL               33: PRIFIL
  8: GLMAS                 21: SODFIL               34: TOPFIL
  9: SOMAS                 22: COMFIL               35: OPTFIL
 10: REFMAS                23: SIFIL                36: CURPRFIL
 11: TYPMAS                24: AIFIL                37: CIDFIL
 12: TAXMAS                25: ARFIL
 13: PRIMAS                26: INFIL

 ENTER DATA SET SELECTION [2                                          ]

CONTINUE |  |  |  |     |  |  | ESCape | MAIN MENU
```

Figure 3 - Data set selection

```
                              DataExpressCenter

  Data fields in MASTER data set PROMAS                       1 of 1

   1:*PRONUM              17: PROARR(11)          33: PROARR(27)
   2: PROTYP              18: PROARR(12)          34: PROARR(28)
   3: DESCR               19: PROARR(13)          35: PROARR(29)
   4: PROUOM              20: PROARR(14)          36: PROARR(30)
   5: PROCST              21: PROARR(15)          37: PROARR(31)
   6: PROLST              22: PROARR(16)          38: PROARR(32)
   7: PROARR(1)           23: PROARR(17)          39: PROARR(33)
   8: PROARR(2)           24: PROARR(18)          40: PROARR(34)
   9: PROARR(3)           25: PROARR(19)          41: PROARR(35)
  10: PROARR(4)           26: PROARR(20)          42: PROARR(36)
  11: PROARR(5)           27: PROARR(21)          43: PROARR(37)
  12: PROARR(6)           28: PROARR(22)          44: PROARR(38)
  13: PROARR(7)           29: PROARR(23)          45: PROARR(39)
  14: PROARR(8)           30: PROARR(24)          46: PROARR(40)
  15: PROARR(9)           31: PROARR(25)          47: PROARR(41)

  ENTER DATA FIELD SELECTION [1,2,3,4,6,7,8,9                          ]

CONTINUE |  |  |  |     |  |  | ESCape | MAIN MENU
```

Figure 4 - Selection of desired fields

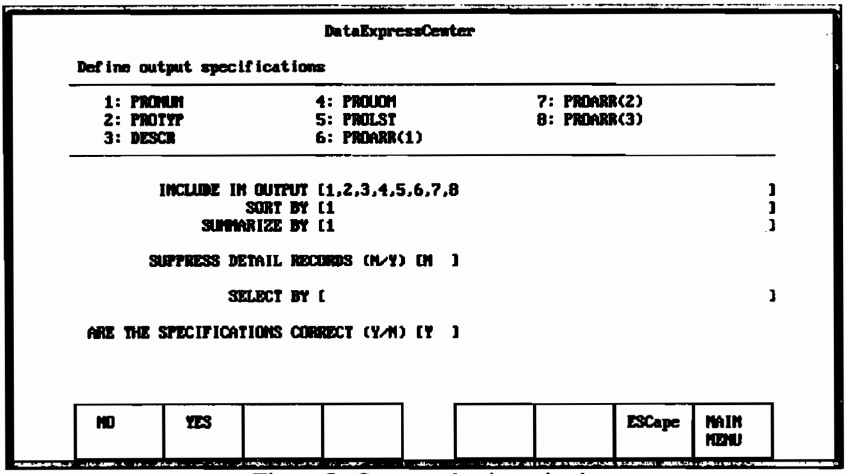


Figure 5 - Output selection criteria

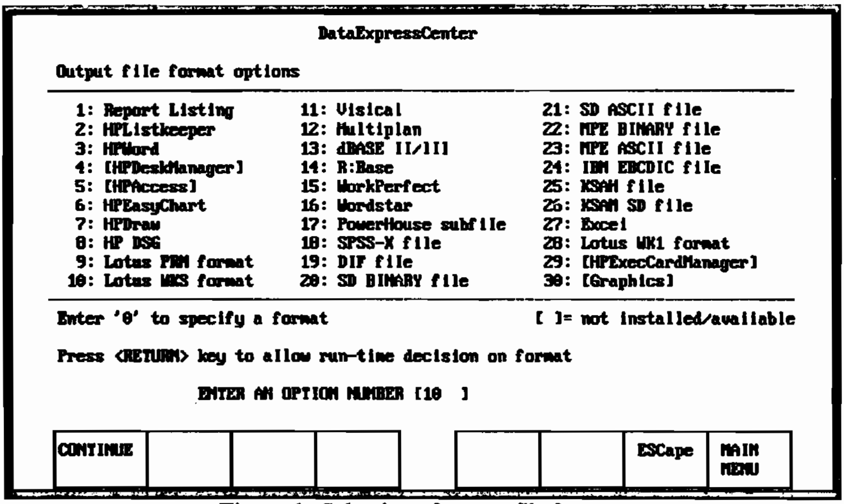


Figure 6 - Selection of output file formats

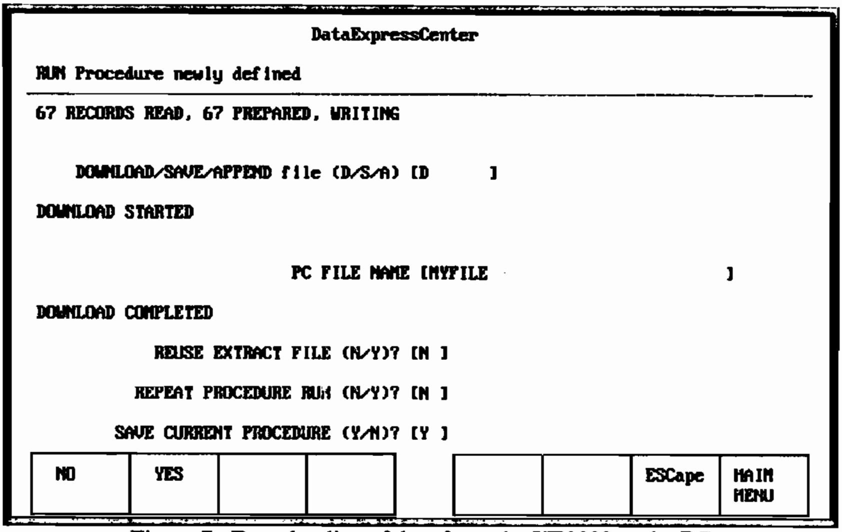


Figure 7 - Downloading of data from the HP3000 to the PC

Step two consists of down loading the information from the HP3000 to the PC. The information can be down loaded to PC immediately after executing the procedure or saved on the HP3000 for transfer to the PC at a later time. Figure 7 illustrates the down loading of the data from the HP3000 to the PC.
Step three consists of loading the information into a PC program. The information down loaded to the PC is now ready to be loaded into an application program such as Lotus 123, Wordstar, etc.

This method offers several advantages over processing the information solely on the HP3000. Once the information is down loaded, the HP3000 is no longer needed. This frees the HP3000 to perform other tasks. The PC user can now manipulate the information in what ever manner is desired. Information can be evaluated in a spreadsheet for "what if" situations, or moved into a database to provide addresses for documents to be printed or reports to be generated. The use or change of the information on the PC does not effect the original version which is located on the HP3000, therefore, data integrity is maintained. It should be understood that the information on the HP3000 may be changing on a regular basis and the information down loaded to the PC may no longer be valid after a period of time. Should the information on the PC be changed in error or lost, a new copy of the information can be obtained from the HP3000. With a program such as Data Express, this procedure can be setup by MIS and then performed by the user without additional assistance from MIS

Off Load Processing consists of tasks performed by the PC but initialized and managed by the HP3000. This method differs from the previous method in that the HP3000 controls the start of the task and the PC to be used. The HP3000 decides the final output destination of the completed task. It may send selected data to several PCs for processing and after the PCs have completed their task, it retrieves the information for consolidation before transmitting to the final user.

The following example illustrates this method.

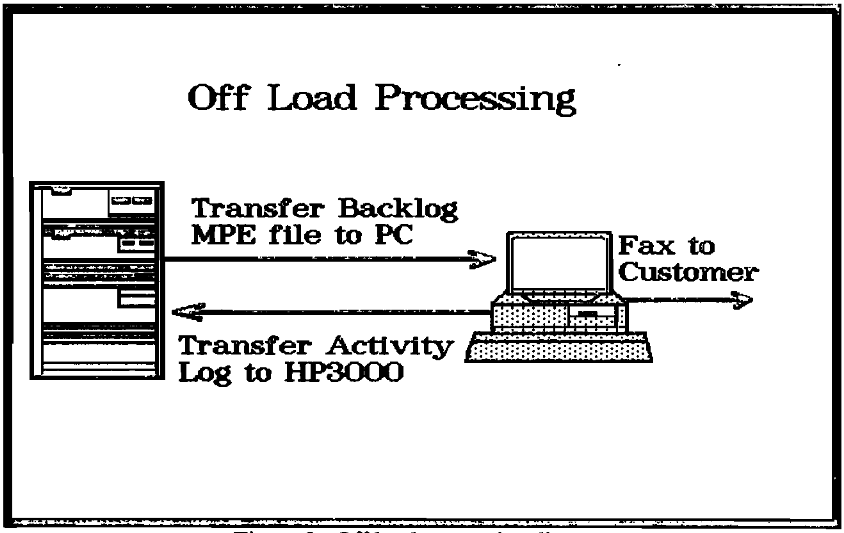


Figure 8 - Off load processing diagram

Example: A problem exists for a manufacturing company in cost effectively informing their customers concerning the backlog of orders. To accomplish this task, it was decided to fax information to each customer on a weekly basis. A PC with Fax hardware and software was used to reduce the load on the HP3000. Figure 8 illustrates this process.

HP3000 schedules a back log report after all orders have been shipped. The back log application program executes, extracts information from the Accounts Receivable database, and stores the information in an MPE ASCII file. After the program completes, a product called HPPCPOLL is used to start a process on a selected PC.

```
HPPCS110                     HPPCPOLL Entry System          02/19/90   19:25
Action:                      PC Header Maintenance           Version 3.00


        PC Name ------------> TEST1
        Logical Device # --> 0022
        Port Speed ---------> 0960
        Modem/Direct Sw. --> M
        Retry Counter ------> 00
        Wait Timer ---------> 0000
        Dial Command -------> 
        ATDT 353 7540


        Last Polled Date --> 02/11/90
        Last Polled Time --> 17:43:42

Enter A(dd), C(hange), D(elete), S(earch), F(ind), N(ext), P(revious)
| END      |PREVIOUS|        |NEXT    |   |PREVIOUS|        |REFRESH|MAIN |
| FUNCTION | FIELD  |        |PC HEAD |   |PC HEAD |        |SCREEN |MENU |
```

Figure 9 - Setup menu for HPPCPOLL

```
HPPCS120                     HPPCPOLL Entry System          02/19/90  20:42
Action:                     Command File Maintenace          Version 3.00

        COMMAND FILE NAME -> FAXINFO

Sw Seq#  Reflection Command, MPE Command, or Comment
 : 0001 CONTINUE
 R 0002 INVOKE C:\FAX\RUNFAX (CMD)


                             File:  C:\FAX\RUNFAX.CMD
                       receive c:\fax\custinfo.dat from custinfo delete
                       shell c:\fax\runfax
                       send c:\fax\logfile.dat to logfile delete

No More Records
Enter A(dd), C(hange), S(earch), F(ind), N(ext), P(revious)
|        |        |        |NEXT    |   |PREVIOUS|NEXT   |PREVIOUS|MAIN |
|        |        |        |COM FILE|   |COM FILE|PAGE   | PAGE   |MENU |
```

Figure 10 - Commands used in off load processing

HPPCPOLL is a product from M.B. Foster Associates Limited designed to control a PC attached to the HP3000. The PC can be attached directly or via a modem. Figure 9 show the setup menu used to control the HP3000 to PC connection. When the HPPCPOLL connection is started, a pre defined series of commands are processed. These commands can be MPE commands, or Reflection commands. Reflection is a terminal emulation package for the PC from Walker, Richter and Quinn, Inc.

Upon completion of the back log program on the HP3000, HPPCPOLL sends a signal to the PC to begin the job of transferring the MPE output file from the HP3000 to the PC. Next, the PC begin the process of merging a cover sheet with the customer information and faxes the complete document to each customer. After all the fax transmissions have been sent, a log of the activity is transferred back to the HP3000 for review by the supervisor. This information can be scheduled to process in the evening to take advantage of the most cost effective telephone rates. Figure 10 shows

Steps in Off Load Processing

1. HP3000 application program generates MPE output file
2. HPPCPOLL begin batch process on PC
3. PC down loads MPE file from HP3000
4. PC Fax program begins fax transmission
5. PC up loads fax activity log to HP3000

Figure 11 - Off load processing steps

the commands used by HPPCPOLL to log on to the HP3000, download the MPE file, and begin the fax process.

In this example, the lengthy process for merging the information with a cover page is handled by the PC. The process is automated by allowing the PC to control the fax of the information, thereby saving on personnel and telephone costs. By designing the software correctly, busy numbers can be re-dialed and a log can be maintained to ensure the activity has been performed completely. Figure 11 summarizes these steps.

This method of improving system performance has several advantages. The HP3000 can schedule this process. If certain programs must be run before the data is ready to be processed by the PC, this can be managed by the HP3000. If a PC is busy or not in service, the HP3000 can elect to perform the process on a backup PC if the backup PC has the necessary capabilities. Several PCs may be used to perform the task in parallel. After all the separate tasks have been performed, the information is transferred to its destination. This may be back to the HP3000 or to a PC for printing.

Share Processing consists of using a PC attached to the HP3000 to provide information to a process running on the HP3000. It frees the HP3000 from the task of searching, storing or retrieving information. This method works the best with historical information that is infrequently used but still needed by a user.

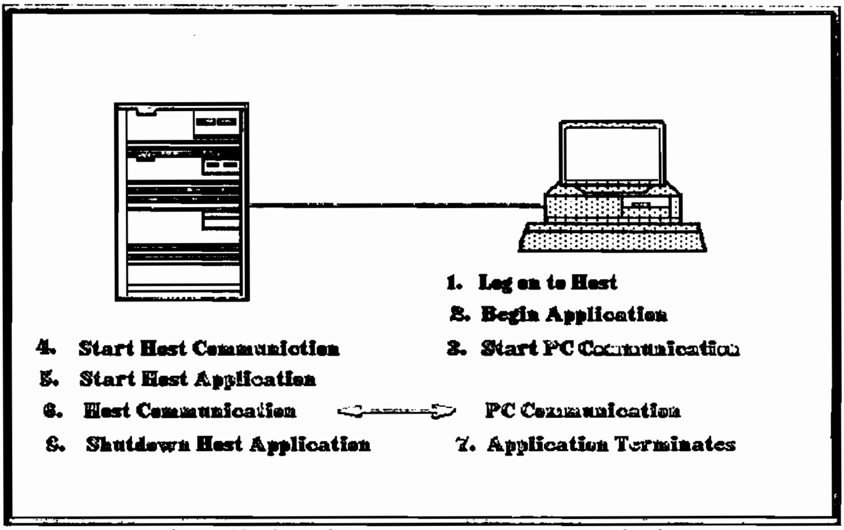


Figure 12 - Steps in process to process communication

Share Processing begins by establishing a communication link between the PC and the HP3000. This is accomplished through a terminal emulation package, such as Reflection, to allow the PC to log on to the HP3000. After a successful log on, the terminal emulation program is terminated. Next the PC application program is started. This program, through a PPL(Process to Process Link) supplied driver initializes the

PPL link. PPL is an software development and communication program developed by Walker, Richter, and Quinn, Inc. PPL allows a process on the HP3000 to communi-

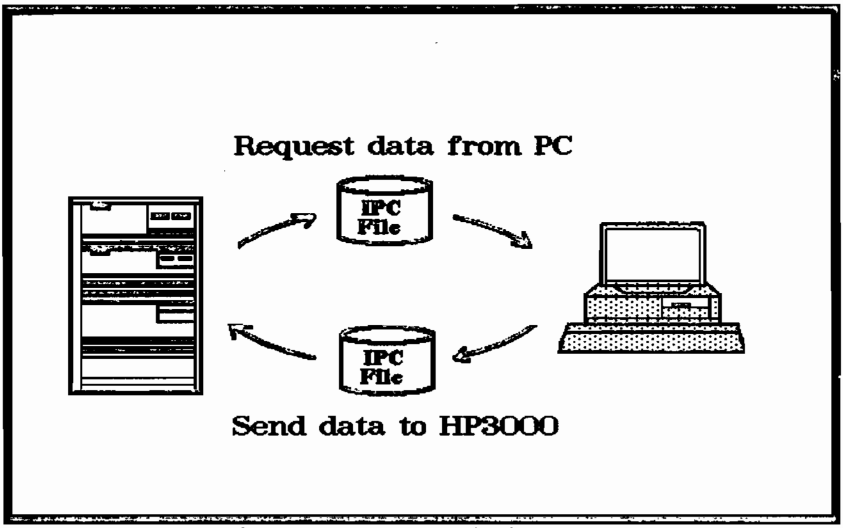


Figure 13 - PPL communication process

cate with a process on the PC. These processes can request or send data in either direction as required. Figure 12 illustrates the steps in a process to process communication link between the HP3000 and the PC.

Once PPL is initialized by the PC application program, it begins a program called Hostlink. Hostlink sets up the Inter Process Communication Files (IPC) to exchange information between the PC application and the host application. PPL is ready to process messages between the PC program and the HP3000 program. Upon receiving a message from the host application program, PPL takes the messages and transfers them to the PC for interpretation. The PC application program interprets the request and performs the required task. The results are transmitted back to the host program by PPL. Figure 13 illustrates the PPL communication process.

Example: Use a PC with an Optical Storage Device as an intelligent controller for storing and retrieving archived information. The present system was designed to interface with the ASK software system, but the concepts can be applied to other archive systems.

A utility program written in "C" on the PC is used to transport records from the

HP3000 to the PC. This is accomplish by using the tool kit included in PPL. The tool kit contains HP3000 Image database calls that can be used by the PC program to find, get, and delete records in an Image database that resides on the HP3000. The records were selected on the basis of status and date. Once the information has been moved from the HP3000 to the PC, the PC is ready to perform its task.

The following steps explain the retrieval process.

1. The PC program starts PPL which creates the IPC files.
2. The PC program waits for a request for information.
3. A user on the HP3000 begins a program to list a document.
4. The user's program checks to see if the IPC files exist. If the files exist, the program indicates the PC is available.
5. The user inputs the document number.
6. The HP3000 application program searches the database on the HP3000 for the requested document.
7. If the document does not exist in the HP3000 database, a request is placed into the IPC file for the document. Requests consist of the following fields: Document_ID, Document_type, Database_number.
8. The PC receives the request and begins the search. If not found, a message is sent to the calling party, "No record found", otherwise the records are gathered and sent to the calling program.

Advantages:

1. Saves HP3000 disk space
2. Allows the PC to manage the data eliminating the overhead normally associated with searching and retrieving by the HP3000
3. Eliminates the need for an operator to restore information from a tape and insert it back into the database
4. Storage capability continues to grow by inserting a new disk

Cooperative Processing is the process of using the PC as a front end processor between the user and the HP3000. To obtain the maximum efficiency from the PC and the HP3000 in an interactive environment, several key issues must be addressed. The issues are as follows:

1. Location and maintenance of information: Data can be stored on the PC, HP3000, or both.
2. Users need to access the information: Edit or view.
3. How often does the information change: Hourly, daily, weekly, monthly, yearly, or not at all.
4. Calculations to be performed: Based on input data, stored data, or both.
5. Screen presentation and cursor positioning.

Synergist, a software development tool from Gateway Systems, Inc., addresses these issues. High volume interactive programs require large amounts of system resources, therefore, care must be taken in dividing the process between the HP3000 and the PC. The following system explains the benefits.

Problem: Develop a means of collecting shop floor information including employee hours at a manufacturing company which is a considerable distance from the HP3000. Insure that system failure does not stop the factory.

To meet the on line real time requirement of a shop floor data collection system, a system must be designed with built in safeguards. This insures that a partial system failure does not stop the factory. In a system that is remote from the HP3000, a failing communication link is a major concern. The ability to capture information and verify input rapidly is a must. The data collection must be easy to use and provide the proper feed back for a user to catch and correct errors before the information is processed.

Synergist provides a development system that allows the designer to distribute the load between the HP3000 and the PC. By providing data synchronization, separate databases can exist on the HP3000 and the PC. These databases can be synchronized on a regular basis, freeing the designer from the complex task of verifying that the information on the HP3000 is in sync with the information on the PC. The data can be synchronized on a data set, record, or field level.

Figure 14 shows the hardware design. A PC network is connected to the HP3000. As input arrives from any one of the PCs on the network, the information is transferred to the server. The server is connected to the HP3000 and verifies the information is correct before updating the database on the HP3000. Static data such as employee ID and employee name can be transferred on a routine basis from the HP3000 to the PC server. This allows continuous verification of employee clock-in and clock-out even if the communication link between the file server and the HP3000 is lost. At the time the communication link is re-established, the transaction received during the communication link down time can be verified and updated. Should an error be detected

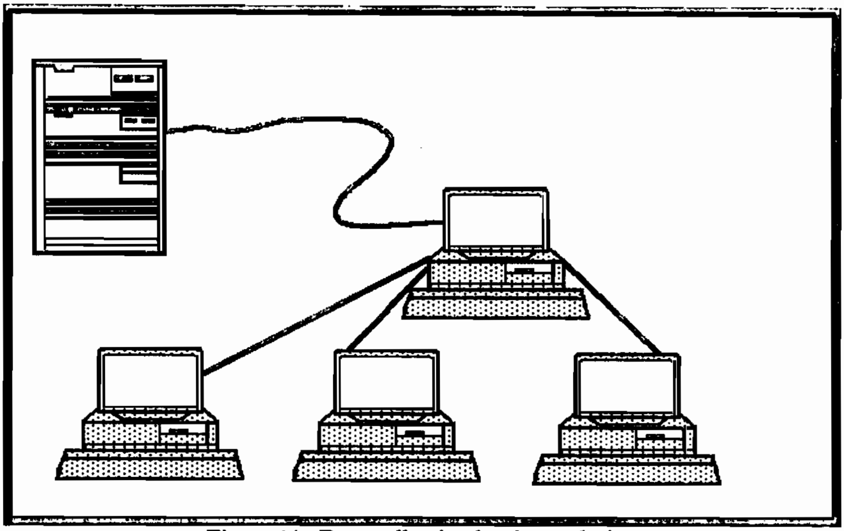


Figure 14 - Data collection hardware design

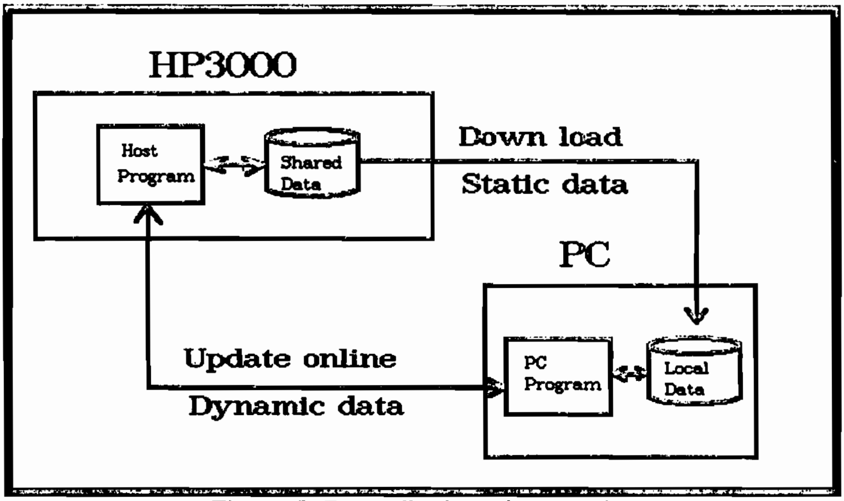


Figure 15 - Data collection software design

in data received during the verification process after communication has been re-established with the host, a flag can be set in the local database which generates a report for the supervisor to correct. By using a PC network, an employee may enter information at the PC that is closes to him. Figure 15 shows the software design.

Development of an interactive system using Synergist, is accomplish on the PC. This reduces the computer resources needed to develop and test the system. The HP3000 is only involved in the testing of the database located on the HP3000.

Parallel Processing is the method by which the HP3000 selects PCs not being utilized to perform selected operations. These operations are performed by a PC instead of the HP3000, much like a math co-processor chip off load mathematical calculation from the main processor. Considerable work needs to be done in this area to perfect this technique. With this method, a system could increase its computing power with the addition of PC instead of an upgrade. An example of this would be the development of a product configurator. A product configurator, which is based on a series of rules, would allow the HP3000 to off load the customer specifications to the PC. The PC would then take the specifications, with the rules of the product, to construct the product's bill of material. The bill of material would be up loaded to the HP3000 for use by the factory.

The growth of the HP3000 and the development of the 386 and 286 computers are only beginning to be utilized in ways that complement each other. These methods and new developments will assist system developers in improving the overall system performance by distributing the load across the computer environment. Distributing the information processing load between the HP3000 and the PC, expresses the idea of the HP3386 and HP3286: The integration of the HP3000 with the 386 and 286 computer.

Acknowledgements

HP is a registered trademark of Hewlett-Packard Company. **Image** and **HP3000,** and **MPE** are registered trademarks of Hewlett-Packard also.

SYNERGIST is a registered trademark of Gateway Systems Corporation.

DATA EXPRESS and **HPPCPOLL** are registered trademarks of M.B. Foster Associates Limited.

REFLECTION and **PPL** are registered trademarks of Walker Richter & Quinn, Inc.

IBM is a registered trademark of International Business Machines Corporation.

80286 and **80386** (286 and 386) are registered trademarks of Intel Corporation.

Lotus 123 is a registered trademark of Lotus Corporation

Wordstar is a registered trademark of Wordstar International, Inc

TITLE: Microcrimes and Megabucks

AUTHOR: Dr. Stanford Sherizen

Data Security Systems, Inc.

5 Keane Terrace

Natick, MA 10760

508-653-7101

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 4084

TITLE: Network Application Development- Integrating PC's and HP 3000's

AUTHOR: Paul Thomas

Crowe, Chizek and Co.

2100 Market Tower, 10 West Market Street

Indianapolis, IN 46204

317-632-8989

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 4088

ORGANIZING INFORMATION THROUGH TECHNOLOGY

Barry R. Prokop
Department of Personnel Services
Fairfax County Public Schools
6815 Edsall Road
Springfield, Virginia 22151

Phone (703) 750-8438
Fax (703) 750-5424

Fairfax County Public Schools (FCPS), the 10th largest public school system in the United States, paid over 25,000 employees in 1989. Understandably, the Department of Personnel Services (DPS) is concerned about its efficiency and effectiveness in dealing with staffing, salary administration, benefits, retirement, human relations, employee relations, certification, and applicant tracking services. The efforts involved in providing these services to FCPS employees and applicants must be coordinated and timely. In order to provide these services the Department of Personnel Services is moving forward in a major automation effort which will transform it into one of the most responsive and proactive Personnel Departments in the United States.

Automation Effort

This automation effort began in 1980 when FCPS developed computer applications on a network comprised of seven Hewlett Packard (HP) 3000 computers to combat the increased complexity of tracking and processing applicant and employee information manually. Significant system enhancements such as the addition of an online applicant tracking system, which enters, tracks, and processes all internal and external applicant requests to include new hires, transfers, and promotion; salary history system which records and maintains an historical record of all employee transactions; retirement system which tracks and generates benefit estimates; evaluation system which tracks teacher performance evaluations and is projected to track all administrative and support evaluations have since been made to the current HP 3000 computer network and are continuing to be made so that the Department can reach its goal of a truly automated system. However, despite these enhancements, the sheer volume of paper and requests was not significantly reduced; ironically, the amount of time to report and research information actually increased.

In 1988, FCPS continued its automation efforts by providing the Department with a technical support team to assist in the coordination of the Department's automation activities. One of the team's first tasks was to review and analyze the Department's processes in order to provide recommendations to improve the Department's effectiveness and efficiency. The team observed that large amounts of administrative information such as policies and regulations, training manuals, and user application support documentation existed within the Department but users could not effectively access and analyze the information because it was either in hard copy format or maintained in different software applications. Since the information was retained in different formats (i.e. hard copy, wordprocessing, spreadsheet, and database) the user's abilities to coordinate the retrieval of specific information requests required more and more administrative time. Each text information request had to be reviewed by the user and all of the information in all of the documents had to be analyzed to access its impact on the request. However, while specific number counts and text searches could be ultimately determined, the overall interrelationships of the texts were not able to be quickly identified because the text information was difficult to evaluate on a global level. Therefore, in order to respond to requests more effectively, the Department decided to review and modify the way in which it handled analysis of large volume documents by creating a model which would facilitate the conversion, maintenance, and access of text information.

Development of Model

In order to create the model, three elements had to be addressed: the conversion of existing documentation within the Department, the implementation of software designed to allow easy collection and retention of text information, and an acceptable delivery system so that users would have access to the information. The end result would be a model that would enable the Department's users to quickly and accurately resolve questions and management requests through access of the information.

One of the key benchmarks of FCPS's success is how quickly and accurately it can respond to questions regarding information for which it is responsible. The inability of any organization, including FCPS, to provide such information can quickly erode confidence in its ability to effectively carry out its mission. Computer access to information has become a critical issue in recent years because it has the potential to transform a reactive organization, which does not have the technological ability to answer requests in a timely manner, to a proactive organization which can access, organize, and report information quickly and accurately.

Before the implementation of computers in the work place most offices within DPS could avoid reporting certain information because it simply required too much time to collect and process manually. Unfortunately, the implementation of computers into the work place has instilled users with the opposite perception: all computerized information is available at a minute's notice.

Additionally, the requirement to respond quickly and accurately has become more demanding in the last few of years because computer hardware technology has increased to a level where it is actually easier and more cost effective to retain large amounts of database information stored on the computer. Since more information is stored in the data bases, it is only natural that increased requests for reports are being solicited by management and users. However, since more information is retained in the data bases it takes longer to find the information necessary in order to make an informed decision. This requirement is not just isolated to FCPS, many other organizations have followed this same

pattern of data processing. For example, the International Association of Universities currently publishes an international catalog which lists all of the universities in the world. Many companies interested in sending employees to universities outside of their native country, use the catalog on a regular basis to locate appropriate universities. The main disadvantage with utilizing this catalog is that it is in excess of 1,000 pages and is sorted by country. For example, if the company wanted to know all of the schools which offered Electrical Engineering degrees, he of she would have to look through each college reference in each country because the text is not structured in a way that allows for flexible searches. Even if the information was converted to computer format, it would still be structured in some type of format that might not be effective for all types of requests.
Another example is when DPS converted all of its manual textbased information to Wordperfect 5.0 formatted documents through the use of an HP scanner and Omnipage text recognition software. The combination of the scanner/software provided the Department with a common technology format upon which to build the model. However, even though the information was stored on the computer in a database, its organization, search, and selection were not greatly facilitated because the software was not flexible enough to allow dynamic searches which could be based on any word or group of words in any document. For example, all of the policies and regulations were scanned from hard copy into WordPerfect 5.0 format and stored as a separate file. Each file contained a separate policy or regulation. A FCPS policy change required that all references to the title "Instructional Aide" be changed to "Instructional Assistant." If the information was stored manually each regulation would be read, however, since all of the documents were in Wordperfect DPS used the search function in Wordperfect to search each of the files, one at a time, to locate each of the occurrences of "Instructional Aide." Even after this was accomplished, additional searches had to be conducted to find "Instructional Aides" and "Aides" or any other type of word choices that referred to this group of people. Another example was when the question was asked, "Is there an proportional number of documented review processes between each of the offices?." To accomplish this task, each of the review processes had to be identified in each of the regulations and then compared by category to each of the other offices. Without a tool to organize the information as well as search for any reference to "review processes," it was still a manually intensive task.

While the capacity to store large amounts of information on a computer through the use of large hard drives, CD-ROM or optical disk drives exists, software products will, nevertheless, have to be used to assist in the organization of this information. To solve this problem, DPS required a software product that allowed information to be collected in a range of formats while providing the flexibility to organize and report in a variety of ways. The most flexible type of software that met this requirement was software referred to as Personal Information Managers (PIMs).

Personal Information Managers (PIMs)

PIMs have been a part of the microcomputer market place for over two years. At first, many software developers placed their products into the PIM category to entice users after Lotus popularized the term with the implementation of their "Agenda" product. However, unlike other spreadsheet, wordprocessing, database, and communication software PIMs did not have a clear definition and purpose. Users quickly became confused concerning the product's focus and were not enticed to purchase it because the definition and purpose of a PIM were never defined or the product's features that were offered overlapped into other software categories. Consequently, the PIM category of products did not gain wide support and recognition in the last two years as the sales figures from many of the companies indicate.

The PIM technology and user market has matured and developed over the last year to a point that generally there are two definitions for establishing a product into the PIM category.

1. Product must be designed to allow the collection of structured (spreadsheets, reports, letters) and unstructured (notes, comments, observations) data from a variety of sources (spreadsheets, databases, wordprocessing).

2. Product must also be designed to organize all of the data from the different sources into categories and relationships.

The main type of PIM that the Department of Personnel Services wanted to utilize was one that could handle large amounts of information from a variety of sources while allowing the flexibility to search by using either the keyword (which is faster) or on any word in the complete text database. Each of the PIM products interprets the PIM category definition in different way and implements it to varying degrees. Some of the products are directed more towards scheduling, administrative tasks, and phone lists. These types of products allow users to enter the independent information in each of the categories and set relationships between the categories. These PIM products should not be confused with Project Management software. Project Management software is structured more for information entry and is not designed for dynamic searches on any part of the entered information. Another type of PIM allows users to track short unstructured pieces of information. The information is entered into the PIM and the relationships are dynamically set up based on the search that is to be conducted. The last type is generally targeted for users who have to handle larger amounts of text oriented data. These PIMs often create their own database or textbase that store the entered or imported information. Although each of the PIMs differ in what category of information is collected and how it is stored, most PIMs allow users to search the information based on preestablished keywords or full text.

IZE

After a thorough evaluation of over 20 PIM products (see appendix for list) the Department of Personnel Services decided to utilize a software product called IZE because it allows the manipulation of large amounts of structured and unstructured information to help reduce the research and analysis time. Some of the projects that the PIM are targeted for include policies and regulations, training manuals, user application support documentation, and research. However, in order to implement the utilization of the software, the text has to be in a format that the PIM software is able to use.

Once the PIM software was selected, the next consideration in creating the model was moving all of the documents into a format that would be easy for users to maintain and utilize by the PIM software.

IZE has the ability to convert many types of software applications, including the software (WordPerfect, 1-2-3, dBASE III, ASCII) that the Department uses. However, since no software is able to handle text that is retained on hard copy, it also has to be converted. To convert this text, the Department used an HP ScanJet + scanner and Omnipage text recognition software. The documents were fed into the scanner through the optional sheet feeder and Omnipage converted the majority of the text documents into WordPerfect 5.0 formatted documents. The scanning was accomplished on an HP ES/12 computer and took about 1.5 minutes per document page. (Increased scanning speed could and will be achieved when the scanner is ported over the new QS/16S computers.) Once all of the documents were converted into WordPerfect format, document maintenance was significantly decreased because online document editing will be available as a result of the software. Even when all of the hard copy documents have been converted, the scanner will still be utilized to convert documents that are received from other organizations or departments that are not automated or do not have compatible software or computer systems.

The last consideration for the creation of the model was the type of delivery system that DPS would utilize in order to provide users with the most current and accurate information textbase. The choices included:

1. Provide a central office of responsibility that would handle the information requests and have the PIM applications stored on one computer. The PIM applications would be stored in one location with all of the requests would be handled by the designated office. The advantages of this approach is that expertise would be quickly developed by that office on the PIM software and its uses and the security of the information updates and changes would reside in one location. The obvious disadvantage to this arrangement is that the volume of requests may actually exceed the available resources to handle them. Additionally, users that are responsible for collecting and maintaining the applications are not held accountable for reporting the information.

2. Provide disk copies of the information bases to individual users in a standalone environment. The distributed information would be controlled by a single user or group of users so that information updates and changes would only be routed through the smallest number of people. The benefits to this type of arrangement are that security risks would be minimized because only a few people would have access to change or update the information. The disadvantage to this type of system is that users would have to receive a copy of an updated disk whenever the texts changed.

Ensuring that each user was working from the most recent copy and that he or she even received the copy would be a time consuming task in a large environment. Additionally, the time required to duplicate a large quantity of disks is both time intensive and expensive.

3. Provide a Local Area Network (LAN) environment which would control access to each of the PIM applications and allow only certain users the capability to update the information. The advantages to this type of arrangement are that the information is distributed to the users who need to access and report on the information. The network and PIM software can control the security to update and maintain the information while allowing the widest exposure to people that may require the information. Additionally, the information in the PIM application would always be current because any update that is completed to the information base is immediately available to all of the users who have access to the information.

Overall, the network environment provides the best delivery system because it allows the most users to access the information while maintaining good security on the integrity of the information. While the network environment is the best delivery system for implementing the PIM applications it should be coordinated with the correct PIM software and has to allow the documents to be easily maintained or converted.

Department of Personnel Model

The Department of Personnel Services has created a model that involves three elements:

1. Conversion of information into a format that is easy to maintain and update.

2. Implementation of PIM software that is flexible enough to handle structured and unstructured data and has enough features to organize, search, and report information based on keywords or any word found in any document.

3. A delivery system that provides the widest opportunity for users to access the information and the best security to ensure that the integrity of the information is maintained.

The PIM software is the key component in the model because it allows users to access, search, and organize information that otherwise would require a large amount of labor intensive work. The ability to search using any word in the text databases or on logical relationships within a text, such as AND or OR conditions, allow better search, analysis, and reporting of the information. The Department of Personnel Services has implemented this model on its policies and regulations and plans to the implement it in training manuals, customer support documentation, research projects in the future.

Attachment

COMPANY NAME	PRODUCT NAME
askSam Systems 119 S. Washington Street P.O Box 1428 Perry, Floridia 32347	askSam
Broderbund Software 17 Paul Drive San Rafael, California 94903	MemoryMate
Brown Bag Software 2155 South Bascom Avenue Suite 114 Campbell California 95008	PC Outline
Chronos Software 555 De Haro Suite 240 San Francisco, California 94107	Who-What-When
Chang Labs 5300 Stevens Creek Blvd. San Jose, California 95129	CAT
Contact Software International Inc. 1625 West Crosby Road, #132 Carrollton, Texas 75006	ACT!
Good Software Corp. 13601 Preston Road Dallas, Texas 75240	Arriba
Group L Corporation 481 Carlisle Drive Herndon, Virginia 22070	Memory Lane
Information Research Corp. 414 East Market Street Charlottesville, Virginia 22901	Syzygy

Attachment (continued)

COMPANY NAME	PRODUCT NAME
Lotus Development Corp. 55 Cambridge Parkway Cambridge, Mass 02142	Agenda
Microlytics, Inc. 1 Tobey Village Office Park Pittsford, NY 14534	Gofer
Micro Logic Corp. 100 Second Street P.O. Box 70 Hackensack, New Jersey 07602	Tornado
Polaris Software 613 West Valley Parkway Suite 323 Escondido, California 92025	PackRat
R&R Associates Inc. 39 Carwell Avenue Mt. Vernon, New York 10552	Shoebox 3
Ten PointO 3885 Bohannon Drive Menlo Park, California 94025	Focal Point II
TimeStar Systems 1803 Mission Street 8 Santa Cruz, California 94039-7210	Active Life
Valor Software 2005 Hamilton Ave. San Jose, California 95125	InfoXL
Persoft Inc. UW Research Park 465 Science Drive Madison, Wisconsin 53711	IZE

Attachment (continued)

COMPANY NAME	PRODUCT NAME
Symantec Corporation 10201 Torre Avenue Cupertino, California 95104	GrandView, More II

PC Programming for Minicomputer Specialists

Russell E. Walker
CompuMed, Inc.
8549 Higuera Street
Culver City CA 90232
(213) 204-4420

Abstract

As PCs continue to proliferate in both technical and business computing, inevitably the need -- or at least the temptation -- will arise to write custom software for the PC. A programming group accustomed to working in a minicomputer environment (HP1000 and/or 3000) will face new issues and constraints in approaching PC development. This paper discusses how to make that transition, drawing on the experience of introducing PC programming to the software engineering group at CompuMed.

Differences affecting software development are discussed in four major categories: hardware, operating system, development tools and the development process. In each area, an overview of the choices and adjustments facing the programming group is presented. Limitations in the typical PC operating system, PC/MS-DOS, probably represent the major hurdle for those accustomed to the full range of services in MPE or RTE.

Of course, the "best" resolution of these issues will be different for each organization and project. The discussion here should help shorten the learning curve for those approaching the PC as terra incognita, and perhaps suggest new options for those who have already blazed their trails.

Introduction

The installed base of PCs now exceeds twenty million (1). Most of those reside in businesses, handling corporate computing tasks right alongside our mainframes and minicomputers. PCs are increasingly replacing terminals on our users' desktops.

PCs are typically used to run a variety of off-the-shelf software packages: word processors, spreadsheets, graphics and desktop publishing packages, among others. Sooner or later, though, the need (or at least the desire) for a custom program or system will arise. When that happens, a programming group accustomed to working on HP3000, 1000 or 9000 systems may be called upon to produce PC software.

Many of the decisions and tradeoffs we make in the software process are based on an intuitive "feel" for the machine and the computing environment, developed over years of experience. That "feel" influences high-level decisions about system architecture and database design, as well as low-level (sometimes subconscious) decisions about the best way to write a section of code. When a programmer, analyst or programming manager moves from a minicomputer to a PC, it is a little like a dancer or an athlete moving from Earth to the Moon: well-trained reflexes can get you into trouble, because they are no longer appropriate for the new environment.

This paper will present an introduction to the PC world from the point of view of an experienced minicomputer specialist. It will explore some of the areas where reflexes may need to be retrained and new approaches may be necessary. Before examining the key differences in the PC environment, though, it is important to consider the reasons for developing custom PC software, as well as ways to avoid doing so by using off-the-shelf software to best advantage.

NOTE: This paper focuses on IBM-compatible PCs, and primarily on the PC/MS-DOS operating system. The reader may attempt to generalize to the Macintosh or other, even more alien environments at his/her own risk.

Reasons to Develop PC Software

Assuming your group develops software for internal company use (rather than for outside sale), and that you have a perfectly good minicomputer, why should you be developing PC software? It is important, when designing a system, to keep in mind the reason you are using the PC as the software platform. Typical reasons include the following:

Improve the User Interface. User's expectations have increased exponentially in this area, as they have become exposed to commercial PC software products with graphical user interfaces, windows, pull-down menus and "point and shoot" operation.

Offload Host Processing. The overall throughput of your information system may be increased by distributing small processing tasks to individual PCs, freeing up the mini for the larger jobs.

Get closer to the hardware. In the simpler PC environment, it is possible to "hook" interrupt vectors, write your own interrupt service routines, and directly control hardware devices on the system bus. The PC is well suited to data acquisition, communications and process-control tasks where being "close to the hardware" is essential.

Use specialized software or peripherals. Many highly specialized software packages and peripheral devices are available for PCs that are either unavailable, or much more expensive, on minis. Bar-code readers, scanners and optical disks (CD-ROM and WORM) are examples.

Allow off-site computing. Having mobile users (such as sales reps on the road) conduct complete sessions via dial-up lines and remote terminals is slow and expensive. Laptop PCs can quickly download the required data and perform processing and reporting off-line.

Set up "cooperative computing." What it all boils down to is that, in the 1990s and beyond, we can no longer assume that "one CPU fits all." The job of a systems architect now includes allocating computing tasks to a variety of CPUs, according to what each does best. This usually means moving toward a "client-server" architecture, in which larger systems fulfill requests from smaller systems, which in turn fulfill requests from their users. Byte magazine columnist Jerry Pournelle's rule of "at least one CPU per user" is increasingly becoming reality, and systems designers must make sure all those computing resources are optimally deployed.

All of the above are valid reasons for allocating processing tasks to the PC, which may in turn lead to a need for custom PC software. One reason that is not valid is the one Mallory gave about Everest: "Because it is there." Programs should not be put on the PC because it is there, or to follow a trend, or because someone on the staff thinks writing PC programs is more fun. The PC and the mini must each be assigned the tasks they are best able to do.

Using Off-the-Shelf Software

Developing custom software, while often necessary, is an expensive and time-consuming proposition. MPE experts are used to using an artful combination of batch jobs, UDCs and utilities like SORT and FCOPY to minimize actual programming. RTE folks do the same thing with command files, and UNIX gurus perform wizardry with shell scripts and a bewildering array of cryptic utilities like "grep" and "uucc".

The same approach can be used on the PC. Before deciding to write custom programs for the PC, explore thoroughly the possibility of using existing tools to accomplish the task, or at least to simplify it.

DOS batch files. If your PCs run the usual PC/MS-DOS operating system, you can build batch files, similar to MPE UDCs or RTE command files, that execute a sequence of commands and/or programs. The PC user enters just the name of the batch file, which causes the whole sequence to be performed.

Parameters can be passed into the batch file from the command line. Limited conditional testing and branching are possible. The execution sequence can be changed depending on whether a file exists, what parameters were entered by the user, and so on. Many DOS commands and utility programs return an "error level" (similar to an MPE JCW) which can be tested in an IF statement and used to modify subsequent execution; enabling, for example, branching to an error section if a COPY or a RENAME command fails.

Third-party utility programs are available that add more capability to batch files. The Norton Utilities is one such package. One of its most useful programs is ASK, which accepts a keypress from the user and returns a corresponding error level code to be tested in an IF statement. This allows dynamically controlling batch file execution based on input from the user.

The DOS batch file facility, at first glance, looks rather limited, but with some creativity it can be "pushed" to do things that do not at first seem possible. The book DOS Power Tools by Somerson (2) gives numerous examples of batch file tricks and techniques, and is recommended reading.

DOS commands and utilities. There is no single command or utility in PC/MS-DOS with the power and flexibility of, say, FCOPY on the HP3000. Nevertheless, the DOS commands XCOPY, FIND and SORT do have capabilities that are not often exploited by casual PC users. Study the DOS manual to become thoroughly familiar with the options and nuances of these very useful commands.

DOS allows "redirection" of standard console input and output for most commands and programs, by using the ">", "<" and ">>" symbols on the command line. This allows tricks similar to using file equations for $STDIN, $STDOUT or $STDLIST under MPE. The output of a command or program, which would normally be sent to the screen, can be saved in a file and used as input by another command or program in a sequence.

DOS also provides a very limited form of a "pipe" (a concept familiar to UNIX pros) for sending the output of one command to the input stream of another. (Actually, a temporary file is created as an intermediate step by the pipe facility.) For example, the file listing produced by a DIR command can be piped to the SORT command to be sorted or to the FIND command to select a subset of files.

Application macros and command languages. Most PC applications have at least a "macro" facility which allows them to memorize and later play back a sequence of commands and keystrokes. Many have built-in command languages that amount to very high level programming languages; these can access sophisticated features of the application and allow reading and writing files, conditional branching, and sometimes even loop constructs.

Reflection, a popular terminal emulator from Walker, Richer and Quinn, has a powerful command language that can handle file transfers to and from the host computer, send commands to the host and interpret responses, read and write files on the PC and perform several DOS commands.

Brief, a widely used programmer's text editor, has a built--in language that can access the search-and-replace functions of the editor to accomplish complex file conversions.

By combining several application programs in series in a DOS batch file, and using the internal command language of each application, fairly complex operations can be automated on the PC. This provides many of the benefits of using fourth generation languages (4GLs), by "leveraging" the capabilities built into each application to handle tasks with a minimum of programming.

For example, a common type of PC/host interaction involves downloading data from the host minicomputer, analyzing it and producing a report on the PC. A DOS batch file might handle such a task by invoking Reflection, Brief and Lotus 1-2-3 in turn. A command file would cause Reflection to log onto the HP3000, run a database extract program, download the resulting file and log off. Brief, executing its own command file, would reformat the data file appropriately to be imported into Lotus 1-2-3. 1-2-3, in turn, would execute a macro to do any required calculations and produce a report. All this would happen automatically in response to a single command (the batch file name) entered by the PC user.

By combining the power of batch files, DOS utilities and application command languages, a great deal can be accomplished without actually writing a line of a code in a third generation language on the PC. There are nevertheless times when nothing will do except to "cut some code" in a 3GL. When driven to this extremity, the project manager, analysts and programmers should be alert to the key differences in the PC environment. These include differences in hardware capabilities, operating system, development tools and the development process itself.

Hardware

Figure 1 symbolizes the major components of a typical PC: the CPU, memory, disk drive(s), keyboard and video display.

CPU. Three classes of CPU chip are widely used in IBM-compatible PCs: the 8088, 80286 and 80386. The 8088 was used in the original PC and PC/XT class machines, and is still found in less-expensive laptops. The 80286 drives AT-class machines, and is probably the most common CPU encountered on corporate desktops. An estimated 40% of all PCs sold last year used the 80286 (3). The 80386 (and its recently introduced successor, the 80486) are usually employed in high-performance PCs used as multiuser systems, network file servers or graphics workstations. Each CPU can run at a variety of clock speeds.

Figure 2 compares published speed ratings, in MIPS (millions of instructions per second), for representative PCs, HP1000s and HP9000s (4, 5). The HP3000 is not included on this chart because, as an "OLTP" (on-line transaction processing) machine, it is not usually rated in MIPS.

Clearly, all PCs are not created equal. It is important to know the mix of hardware that your PC application must run on, before designing the application. A graphical user interface or a complex algorithm that flies on a 20-MHz 80386, may be tolerable on a 12-MHz 80286 but glacially slow on a 4.77-MHz 8088.

Another important variable is the presence or absence of a numeric coprocessor. Each of the CPUs can be augmented by an optional coprocessor chip for floating-point math operations. If your application involves extensive floating-point operations, especially geometric calculations, you should find out whether your target hardware includes a coprocessor. (The coprocessor is irrelevant to integer math and string operations.)

Memory. PCs can be configured with varying amounts of random access memory (RAM), ranging from 256K bytes up to several megabytes. Until fairly recently, the most common configuration for business PCs was either 512K or 640K. Now, with RAM prices falling and applications expanding, 1M, 2M and 4M machines are increasingly common.

FIGURE 1: PC COMPONENTS

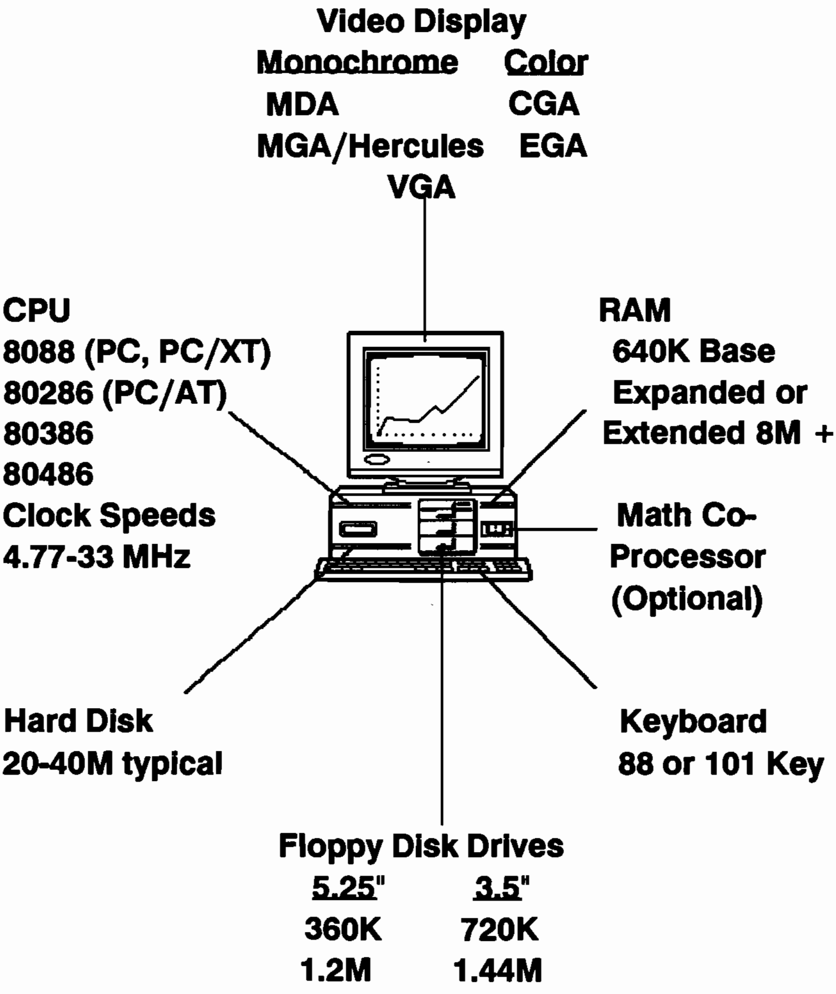
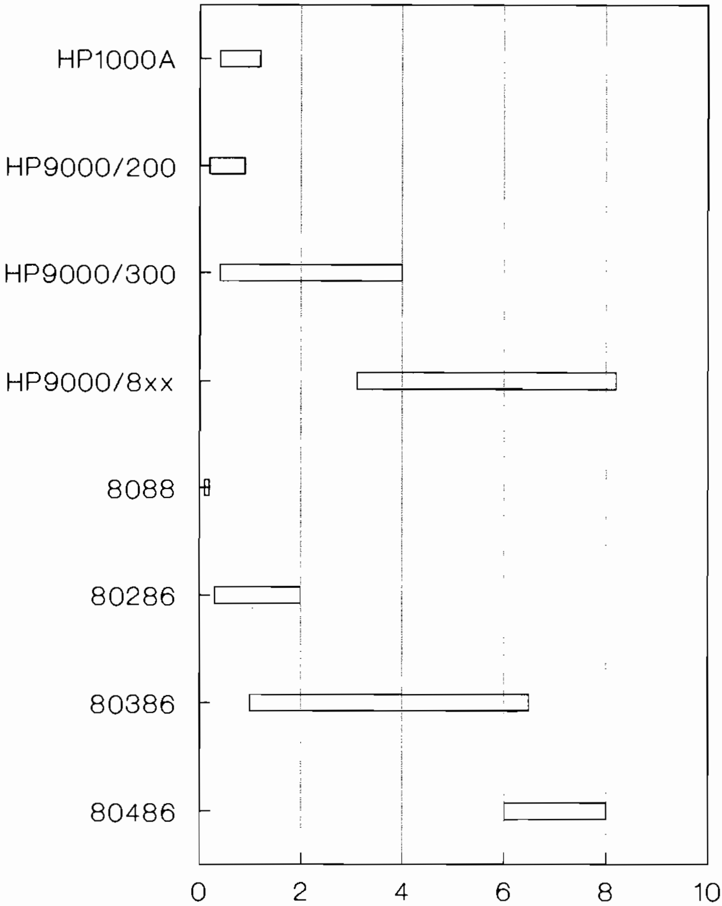


Figure 2: Processor Speed (MIPS)



Under the MS/PC-DOS operating system, only the lower 640K of memory is readily accessible to applications, even if the machine contains several megabytes of RAM. Memory beynd 640K is considered either "extended" or "expanded" memory, meaning it can be accessed only by putting the CPU into a special mode or via a bank-switching scheme. In practice, this can mean that the extra memory is used only as a RAM disk. Special drivers and programming techniques must be employed to use this extra memory for data storage within a program. Access to memory beyond 640K is easiest with the 80386 CPU, which has built-in memory management.

The readily accessible 640K can get crowded surprisingly quickly. Application programs must share the memory space with the DOS operating system, device drivers, various stacks and buffers, and any "terminate and stay resident" (TSR) programs (like Borland's Sidekick) that hang around in the background, using up memory, even while not executing.

Minicomputer application programmers are not usually conscious of memory constraints, encountering them only when working on very large programs. Minicomputer operating systems shield applications programs from these constraints with sophisticated memory management techniques. On the HP3000 under MPE, for example, virtual memory for code is provided; code segments are brought into physical memory from disc as needed. Program segmentation is easily accomplished using features built into the high-level languages and supported by the operating system.

It is possible to create PC programs which use overlays that are dynamically brought into memory from disk. However, under PC/MS-DOS, this requires using third-party utilities and special programming techniques. Virtual memory is not supported by DOS itself. Coupled with the inefficiency of PC compilers in code generation, this makes memory constraints on the PC of much greater concern.

For any program of significant size, it is advisable early in the development process to produce a reliable memory usage estimate. Create and compile representative samples of code, and extrapolate to the size of the completed program. Add any significant data structures, file I/O buffers, etc., plus the program stack. If this produces an estimate of 400K or greater, consider breaking the program into two or more smaller programs which execute sequentially.

Disk. Mini and PC people don't even spell the word for "rotating auxilliary data storage media" the same way: it's usually "disc" for minicomputers and "disk" (short for "diskette") for PCs. Most desktop PCs sport hard disks with capacity in the 20-40M byte range. File servers may have much larger disks, 300M or more.

PC disk storage is not nearly as robust as minicomputer disc storage. This means that more attention must be paid in a system design to assuring data integrity and recovering from disk errors. Although it is not good practice, one can often get away with not checking file I/O error codes on minicomputers. This will almost always cause trouble on the PC, usually sooner rather than later. Special programming effort may be required to "trap" disk errors in the application program and handle them in a reasonable way. (DOS's default response to a file error is to give the user a cryptic message asking "Abort, Retry, Ignore, Fail?" All these choices are bad.)

Video Display. There is a plethora of PC video "standards" -- MDA, CGA, Hercules/MGA, EGA, VGA, "Super" VGA, etc. It is essential to use a good third-party library package to insulate the application program from the details of handling all these incompatible displays.

The serial connection between a terminal and a host computer is a data bottleneck. Even with a direct connection at 19.2K bps, repainting a complex screen on a terminal creates a noticeable delay. PCs use a memory-mapped video display, so data are transferred to the screen at memory-update speed. This permits the development of complex windowed displays. The user interface design should take advantage of this capability, since it is one of the PC's strong suits.

Operating Systems

In the world of HP minicomputers, the system you work on defines the operating system you use: RTE for the 1000, MPE for the 3000, UNIX for the 9000. Many PC operating systems exist, although not all are available or practical on low-end hardware configurations. If you are developing a stand-alone application, you may have the freedom to pick an operating system best suited to the project requirements. On the other hand, you may be locked into the operating system in use on your organizations's installed base of PCs.

PC/MS-DOS. This is by far the most popular PC operating system. It will run on practically any hardware configuration. It is also the most limited from an application developer's standpoint. However, it is well suited to a relatively straightforward programming style, and a wide variety of development tools will run under it.

One of DOS's worst shortcomings, as discussed earlier, is its nonexistent support of memory beyond 640K bytes. This is a severe limitation for the development of large systems.

DOS is a single-user/single-tasking operating system. A DOS program cannot spawn a subprocess or stream a batch job to perform background processing. This can be a problem for an application which must handle device I/O or do calculations while continuing to respond to user inputs.

The services DOS provides to application programs are fairly meager, compared to the array of system calls available to the MPE, RTE or UNIX programmer. Since multitasking is not supported, there are no interprogram communications facilities. Serial port I/O is so poorly supported that the DOS calls for this are almost universally ignored. Only very basic video I/O services are provided, and these are often bypassed for the sake of speed. Printer drivers are not available at the system level, so each application must include built-in support for the unique features of all the printers it will work with.

Many of these deficiencies can be remedied through library routines supplied with the programming language or as part of a third-party library. This increases the chance that features will be implemented differently from one program to the next, and makes the choice of a standard language, compiler and library set all-important.

DOS Add-On Packages. These are software systems that operate in between PC/MS-DOS and application programs, providing capabilities that DOS alone cannot. The most commonly used are Quarterdeck Office System's DESQview and Microsoft Windows.

DESQview creates a multitasking environment within the normally single-tasking world of DOS. Multiple DOS programs can be run in DESQview windows and timesliced. DESQview also provides some relief from DOS's memory-management problems, since it will suspend programs and swap them out to disk to free memory for other programs. A special version for 80386-based PCs allows several programs totalling more than 640K to be active at the same time (although no single program can be larger than 640K).

The most recent versions of DESQview include a set of Application Program Interface (API) routines to be used in programs specifically engineered to run under DESQview. The API routines supply capabilities like interprocess communication and video screen management. A "panel design" tool constructs predefined screens for use within programs (similar in concept to VIEW/3000).

Microsoft Windows also allows multitasking on the PC. Windows provides tools for building complex yet standardized graphical user interfaces. Many PC users are already familiar with the Windows conventions of pull-down menus, scroll bars and dialogue boxes, so developing software under Windows (and adhering to the conventions) may shorten the learning curve for your users.

Windows programs incur fairly heavy penalties in memory and processing overhead. A fast 80286 system or, ideally, an 80386 is required to run Windows at an acceptable speed. Programming for Windows requires a radically different style than minicomputer programming; Windows programs basically respond to external events (such as the user clicking a mouse button on a menu choice) rather than following their own agenda and demanding a user's input when it's needed. This can mean a long learning curve for designers and programmers who are "doing Windows" for the first time.

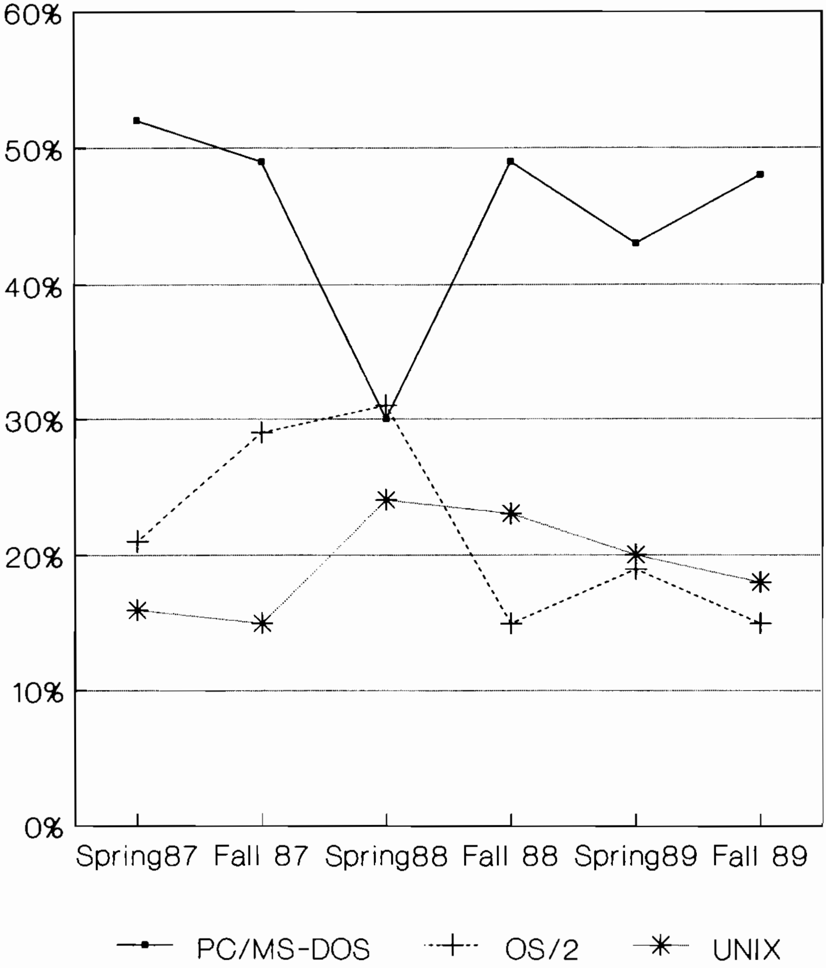
<u>Alternatives to PC/MS-DOS.</u> If your application does not have to fit a Procrustean bed of installed DOS machines, there are several alternative operating systems to consider. The most "mainstream" are UNIX/XENIX and OS/2.

Several varieties of UNIX are available for PCs. UNIX provides robust multitasking, virtual memory management and an excellent development environment. It requires a high-end PC hardware configuration -- at least an 80286 CPU, a large hard disk and several megabytes of RAM. UNIX systems, even on PCs, require expert setup and system administration, and have a reputation (mostly deserved) for being "user-unfriendly" to nontechnical users. UNIX on a PC will probably make HP9000 pros feel right at home, and is a good choice for heavy-duty standalone or multi-user applications.

OS/2 also provides industrial-strength multitasking and memory management. It includes a Windows-like graphical user interface called Presentation Manager (PM). Like Windows, PM requires minicomputer programmers to adopt an unfamiliar programming style. OS/2 may be well suited for a dedicated application but has not found wide acceptance on desktop PCs because of a dearth of off-the-shelf applications software (although that is slowly changing). Like UNIX, OS/2 requires a high-end PC hardware configuration.

To put all this in perspective, Figure 3 shows the results of a poll conducted by <u>Byte</u> magazine at recent Comdex computer shows (6). PC/MS-DOS remains the "dominant" operating system for PCs, and is expected to remain so for some time to come.

Figure 3: COMDEX Poll Results
"Which OS will dominate in the future?"



(Polls conducted by BYTE magazine.)

Development Tools

As with operating systems, the key word for PC development tools is "choice", or perhaps, to use Toffler's word, "overchoice."

Language. Underlying all the other development choices is the selection of the programming language. Virtually any language dreamed up by anyone, anywhere, is available in some version on the PC platform.

One obvious choice is to use the language you are using now on your minicomputer. Worthy versions of FORTRAN and COBOL are available for the PC. I would recommend this choice if chunks of existing code can be ported to the PC, or if your project is too small to justify the learning curve of a new language for the programming staff.

In most cases, though, I would suggest that you bite the bullet and develop your PC applications in the C language. There are two reasons for this. First, since C is used by most developers of commercial PC software, more third-party function libraries are available to support this language. Second, developing in a different language on the PC will emphasize to the programmer that he/she is working in a new environment, and help avoid falling into inappropriate old programming habits.

Compiler. Again, this is an area where you may not be used to having a choice. If you program in COBOL on the HP3000, chances are you use HP's COBOL compiler. On the PC, you have a variety of compilers to choose from for most popular languages.

Assuming you are working in C, I would recommend one of the "mainstream" compilers, either Microsoft C or Borland Turbo C. With one of these, compatibility with nearly all third-party function libaries is assured. (CompuMed standardized on Lattice C several years ago, because of the availability of a compatible Z80 cross compiler for a specialized application. We have not so far had a reason to change.)

Some C development packages, notably Turbo C and Microsoft Quick C, feature an "integrated environment" combining editing, compiling, linking and debugging. Although this may be a productivity booster for some, many minicomputer veterans may be more comfortable with the traditional command-line interface and separate editor, compiler, linker and debugger.

Function Libraries. The key to successful, productive PC software projects is full use of the many available third-party function libraries. There are hundreds, if not thousands, available, and they can save many man-hours of programming by avoiding reinventing the wheel.

At the outset of a PC development project, pick one or two comprehensive general-purpose libraries, plus any specialized libraries needed for data communications, graphics, etc. We have used the Greenleaf Functions and Essential C libraries to good advantage. Greenleaf is particularly recommended because, in addition to a general library, they offer several special-purpose libraries. Mixing and matching function libraries from different vendors, although possible, is somewhat risky since there can be naming conflicts or other incompatibilities.

Editor. A full-screen programming text editor is a must. There are many excellent editors for the PC, and choosing one is largely a matter of taste. If possible, obtain demo copies of several editors, let the programming staff try them out, and pick a consensus favorite.

CompuMed has used the Brief editor for several years with good results. Early versions used nonstandard keystrokes for certain functions, which made changing to other editors or word processors awkward, but later versions have mostly fixed this problem. A major advantage of this editor is its very powerful built-in command language, which can automate complex file conversions.

Several compiler packages include editors (for example, Lattice C includes LSE, the Lattice Screen Editor). Although usually not as full-featured as standalone editors like Brief, these are usually well integrated with other compiler utilities and may suit your needs and preferences.

Code Librarian Software. For all but the smallest projects, a code librarian or version control system is recommended. These utilities organize the project's source code into modules, keep track of revision levels, and allow programmers to check modules in and out of the library for work. Usually, a UNIX-like "MAKE" capability is provided to automatically rebuild any specified revision level of the program. These utilities are nearly essential for keeping track of a project's source code, when each programmer might have his/her own version of each routine on his/her own PC's hard disk.

There are many good librarian programs. In this case, probably the project manager should choose a librarian system that suits his/her project management style, in concert with the senior staff member who will be responsible for final integration and testing.

Computer Museum

The Development Process

During a minicomputer software project, all programmers work on the same machine and automatically share the same development tools and environment. The software will run in production on the same machine or a very similar one. Such is usually not the case for a PC project, which creates some interesting differences in the development process.

The Need for Standards. It is critical that the project manager establish and enforce standards for the entire team as to which compiler, utilities and function libraries are to be used. The programming team must use a common development environment, even though they are working on separate machines. It is also important to establish and use the same tools from one project to the next, so the organization does not end up with a hodgepodge of software that is a maintenance nightmare.

It is also important to establish a standard PC directory structure for the project, which will be set up on each team member's PC. This is critical so that, when a team member is absent or leaves the project unexpectedly, others can find their way around on his/her hard disk.

It is possible to allow some flexibility in each member's choice of editor, provided all the editors generate ASCII files acceptable to the compiler and have the minimum set of features required for the project. I would tend to encourage standardization even here, to strengthen the team concept of shared goals and a shared environment.

Establishing and enforcing project standards may meet with a surprising amount of resistance, especially from team members with PC experience. The first word in PC is "personal", and users tend to get territorial about "their" PCs. This is a new phenomenon for the minicomputer manager, who is probably not used to programmers wanting to bring in their own HP3000 COBOL compiler from home. The best approach is to involve the team members up front in determining the project standards, then enforce them rigorously and hang anyone who persists in operating outside the agreed-on system.

If at all possible, all team members should have identical PC hardware configurations for development. You will eventually want to make sure that your software will run properly on all the hardware variations it may be exposed to; however, this is a needless complication during development and initial testing, when confidence in the code is low and each problem has many possible causes.

Version Control. This becomes more of a potential problem on PCs because several inconsistent versions of the same code can exist on different team members' hard disks.

To maintain configuration control, during initial development, let each programmer do coding and unit-testing on his/her own machine. As each unit is tested, it should be signed off by the programmer and impounded by the project manager. One machine, under the control of the manager and/or the senior programmer who will do system integration, should be designated to hold the impounded code, which will be managed by the code librarian utility.

Some code units, such as header files and utility functions, will be required by other team members. Updated versions of these units should be distributed on a regular basis, so that no one is working with an obsolete version. Pre-release revision levels should be updated and tracked rigorously throughout the project.

This whole process is made much easier if the development team's PCs are networked together, so code can be transferred merely by copying it to or from a designated directory on the file server. Security provisions in the code librarian can prevent unauthorized modification of impounded code. The process can also be managed using floppy disks for file exchange, though this is somewhat cumbersome and requires more discipline to work properly.

System Integration. One team member must be designated to perform the final system integration and testing, on a PC designated for this purpose.

Before testing, the software should be rebuilt by the integrator from the "final" version of source code stored by the code librarian. (The system-building process should be automated since it will probably be performed several times during integration and testing.) When problems show up, the integrator should identify the offending module and return it to the appropriate programmer for corrections. When fixed, the module should be checked back into the code library, the system rebuilt again, and integration continued.

System Testing. Most organizations have accumulated a mixture of PC hardware configurations. Although it may not be possible to exhaustively test a system on all possible hardware combinations, it should be tried with each major hardware element before being released, to detect gross incompatibilities. This is particularly important with respect to the video display, since a clear, easy-to-read screen on one display may turn into a garish mess or a uniform grey muddle on others.

The ideal way to do this is to set up a test bed system. Start with the integration PC's configuration, with which the software is known to work. Then change one hardware element at a time, in a controlled fashion. A less formal approach is simply to test the application on a representative sample of PCs around the organization. This will reveal any obvious problems, but may make it harder to identify their cause.

Applications should be tried on a range of CPU types and speeds. Development PCs are usually faster than the typical business desktop machine; this may mask serious performance problems.

It is important to set up a testing regime that adequately samples the PC variations in your organization. Compatibility problems will be much easier to solve during prerelase testing than weeks or months later, when a user happens to stumble upon them.

Conclusions

A minicomputer software group can develop PC applications, but some readjustment of the group's mind set is required for the new environment. In many cases, batch files, PC utilities and the command language features of off-the-shelf software can be used to eliminate or minimize the need for custom programming.

The PC environment, for both development and execution, is much more heterogeneous than minicomputer veterans are accustomed to. Wide variations in PC hardware configurations, and their impact on application design and performance, must be understood. A choice of operating systems is available; PC/MS-DOS, still the dominant PC operating system, has severe limitations but can be augmented with add-on packages like DESQview or Windows. Development tools can be selected to fit project requirements and the preferences of the programming team. Project managers can cope with diversity by defining a standard development environment, controlling code versions and planning for careful system integration and testing.

You, too, can develop PC applications for fun, profit and career advancement. Happy programming!

Note

All product names mentioned in this article are trademarks of the associated companies. Specific products are discussed for information purposes or as examples, and are not endorsed by the author or CompuMed, Inc. The author is not affiliated with the companies or products mentioned, except as an end user.

References

1. Heller, Martin, "Redefining the Standards", Byte, Fall 1989 (IBM Special Edition).

2. Somerson, Paul, DOS Power Tools: Techniques, Tricks and Utilities (New York: Bantam Books, 1988).

3. Hayes, Frank, "The Spirit of '86s", Byte, March 1990.

4. Campbell, Gordon A., "Inventing the PC's Future", Byte, Fall 1989 (IBM Special Edition).

5. ---, Hewlett Packard Test and Measurement Catalog 1988.

6. Ryan, Bob, "The Succession Crisis", Byte, March 1990.

PC/HP3000 Integration: Who Does What?

Craig Fransen
Fransen/King, Ltd.
16400 Southcenter Parkway
Seattle, Washington 98188
(206) 575-1570

Personal computers (PC's) are rapidly becoming the method of choice by which end users are connected to HP3000 computers. The power of PC hardware and software seems to increase exponentially with each passing year and making the most of this new-found power has become a priority in many organizations. Many software firms are working to address these needs. This paper surveys these efforts and explores the state of PC/HP3000 integration now and for the future.

Where We Have Been

The history of PC/HP3000 integration provides a basis for predicting its future. In the past, the CRT display terminal, in combination with MPE programs, provided (and dictated) all of the flexibility and intelligence in manipulating the data resources present on the HP3000. These programs, for the most part, were not written by end users but rather developed by third-party vendors or in-house DP departments.

As such, the end users were always a "step away" from the data. They were forced to communicate their needs to analysts and programmers who then either developed or purchased applications to fill those needs. This process was affected by schedules and budgets which caused a lag between users articulating a need and the development of a system to fill the need. A change was needed.

"Data Base" Concept

Long before the PC's introduction, another change was occurring. End users and DP departments realized that much of the information concerning the organization was interrelated.

It seems obvious today that in many organizations, financial data is related to sales data which is in turn related to order taking and manufacturing data. And yet in some older systems, information concerning a particular customer, for example, might exist in many different files and formats on the system.

To address this problem, the concept of an organizational "data base" was born (the term "data base" is used here in its general sense - not to the HP3000's IMAGE data base system in particular). In its simplest form the definition of this concept is that a particular piece of information - say a customer's address - exists in one and only one place on the system. Any process needing that piece of information looks in that place to get it.

In adhering to the data base concept, it became possible for the first time to ensure that the information reported in one process or system would match that reported in another.

A major conceptual change had occurred: from the user's point of view, the HP3000 had simply become another PC application!

Redundancy was eliminated saving disc space and headaches. Control was enhanced as information concerning the entire organization was put into one form - not spread among vaguely connected files.

The end users, however, were still a step away (albeit a shorter one) from the data.

The Advent of the Personal Computer

In the early 1980's, PC's began to make appearances on the desks of end users. At first it was not entirely clear how, if at all, these machines affected the organization's DP strategy. The applications which sprang up on these machines were for the most part either unavailable in the same form on the host (e.g.; spreadsheets, graphics packages, etc.) or clearly superior for some users on the PC (e.g.; word processing).

For some users, PC's represented a rebellion against an unresponsive centralized DP bureaucracy. They wanted ownership and control over "their" data. They maintained that just because the "C" in PC stood for "computer", it did not automatically follow that these new machines came under the purview of data processing.

In some cases, data processing departments were only too happy to have a reduction in the number of end users pestering them for small applications. Organizational attempts at control over these proliferating powerhouses consisted mainly of standardizing on a particular brand of hardware and/or software. Most of these attempts failed in the end as users purchased the latest whiz-bang hardware and software anyway.

Organizational data began to be spread among these PC's. The data were being manipulated by spreadsheets and other programs created by end users. The output from these systems was being used in the management of the organization. The data bases of these organizations were dissolving.

In addition, the practical controls developed and imposed over the years on the DP department's systems were nonexistent in their PC cousins. Spreadsheet output, printed on near-typeset quality printers, looked wonderful - but were the numbers correct? How does one even approach the verification of the correctness of a tangled web of spreadsheet calculation?

PC disk drives began to sport twenty, thirty, even eighty megabytes of data which was backed up rarely if ever. Peripheral devices such as printers and plotters of varying stripes were purchased and connected. Mayhem seemed at hand.

PC Integration Begins

In addition to myriads of stand-alone PC applications, some users required access to the HP3000. Terminal emulators were introduced which seemed to take a giant leap backwards turning these new powerful PC's back into simple terminals. In fact, however, a major conceptual change had occurred: from the user's point of view the HP3000 had simply become another PC application!

A clear separation still existed, however, between the host HP3000 and PC. HP3000 data and programs were separate and in a form different from PC data and programs. PC's could not utilize HP3000 peripherals nor could HP3000 applications utilize PC peripherals. At first, the PC's display screen and keyboard were the only things "connected" to the HP3000. The connection needed to evolve.

Terminal emulators began to offer "file transfer". With this facility users could transfer files from the PC to the HP3000 and back. Simple sharing of data had begun. PC printing utilities were introduced which allowed PC applications to print to HP3000 spooled printers. Simple peripheral sharing had begun and the HP3000 had regained its status as a central connectivity point. The DP department was forced to look up from their coding sheets and start to support these new users on their system; and while no one was looking, the PC users were being brought back into a new DP fold.

New Possibilities

Now, the process is continuing. New software allows PC to use all the major peripherals on the HP3000 - disk drives, tape drives, and printers. HP3000 applications can spool output to PC printers.

The next step is the integration of PC and HP3000 applications. These new integrated systems let each component do what it is best at doing.

The PC forms the user/machine interface as well as providing significant local processing. The HP3000 provides a powerful central hub.

Unlike other network servers whose operating systems have been modified to provide "file sharing", the HP3000 was designed from the ground up to perform multitasking and effortlessly shares data between any number of users. Bringing the PC applications (and users) back into the sphere of data processing re-establishes the "data base" of an organization's data described earlier.

By making the data easily accessible, PC users are no longer tempted to create private copies of data with the problems that act engenders. HP3000 applications which formerly were limited to those display and data manipulation facilities available on the host can now offload much of that processing to PC's.

In the area of control, this integration and the possibility of implementing a "one and only one copy" or "data base" concept of data storage brings obvious benefits. Although, as we will see later, the data base concept does not imply a single central data repository, it does imply that a single "logical" data base exists - even though spread throughout a network.

PC software distribution can now be handled in a centralized fashion with new and upgraded software "downloaded" automatically to connected PC's.

Spreadsheets and other PC programs can now be scrutinized for correctness and distributed for use throughout an organization.

Local data storage on a PC can be routinely backed up to the HP3000 host.

End users get the best of both worlds. By using PC software designed specifically to be easy for non-DP people to use connected to HP3000 computers designed to share information, they need no longer be a "step away" from the data.

The State of the Industry

With product announcement news breaking daily, any snapshot of the state of the PC/HP3000 market is inevitably out of date. Accepting that caveat, we can take a look at the products available today, those which will be available "real soon now", and, perhaps, muse upon those coming in the future.

It is useful to divide the market into three major categories: peripheral sharing, data extraction/download, and products implementing "cooperative computing". All of these, to some extent, depend upon both the HP3000 and the PC to perform their functions.

Note that communications strategies utilized by these systems include both serial (RS-232/422) and LAN-based paths. Most are currently serial, some both, some LAN only, but with the increasing availability of various LAN connections into the HP3000, most will be offering such connections in the future.

Briefly, communication between machines is usually viewed as a series of "layers" present on each machine, each of which is responsible for a particular activity. The lowest layer actually moves the data from machine to machine, while higher layers communicate with these lower layers to provide some particular service to PC application programs.

For the most part, these products form layers above the lowest datacomm layer, and, if properly written, can be made independent of the lower "transport" layers and thereby utilize various current and future datacomm paths into the HP3000 host.

Peripheral sharing is perhaps the most straightforward of these capabilities. Packages implementing peripheral sharing include Resource Sharing, Office Extend, RSVP, AdvancePrint, and Minisoft 2392. All of these systems allow PC's to utilize spooled host printers for output from PC based application programs. For any peripheral capable of being spooled off the HP3000, these packages provide an organization with a way to spread the cost of expensive printers and plotters among many PC users.

Two of these systems (Resource Sharing and Office Extend) also allow PC's to utilize HP3000 disk space as if it were a local PC disk drive. Turning the HP3000 into what is known as a "file server", these systems allow data files to be shared among PC users. Many modern PC application packages can utilize the networking features present in MS-DOS versions 3.x and later to coordinate access to these files.

Both of these systems also provide a mechanism to back up local PC disk drives to the HP3000 tape drive. One of them (Office Extend) also allows HP3000 spooled output to be routed to printers attached to the PC itself.

Data extraction and download packages provide a mechanism to move data present in various forms on the HP3000 host into PC based application packages. The most common format provided is Lotus 1-2-3 worksheet file format and the systems can variously convert data from IMAGE, KSAM, SD, and "flat" MPE files as well as from printed (spooled) reports into a form which can be either retrieved or "imported" into Lotus.

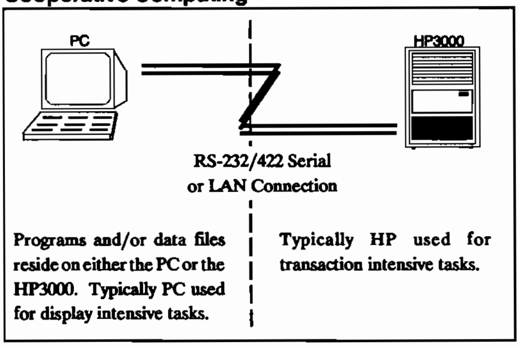
Packages implementing this capability in general include SheetMate, DataExpress, SpeedDoc, GUS/3000, Visimage, and OmniView. NewWave, in conjunction with Information Access, can also be viewed as performing this function.

Some also can produce data in other PC formats including Excel, DBase, DIF, comma delimited ASCII, etc. (DataExpress, Visimage).

Two of them (DeskDirect and SpeedDoc) utilize downloaded information as part of their processing: IMAGE, KSAM, and MPE file data to be included as part of a word processing document (SpeedDoc), PC and MPE files sent or received via HPDesk (DeskDirect).

Leaving the most exciting for last, we come to Cooperative Computing. Products in this category provide the capability to distribute computing on both the HP3000 and the PC to perform a specific function. Some of these packages go further by providing the capability to distribute general computing.

Cooperative Computing



Products implementing specific functions include AdvanceMail and Dispatch, both "front ends" for HPDesk.

More general program development capability is offered by Oracle, Powerhouse, Synergist, and Speedware. These systems provide mechanisms to spread both data and computing between the HP and the PC.

Products offering program to program communication which can be utilized by custom applications include PPL and Office Extend.

The Future

The trend in PC/HP3000 integration seems to be toward a sort of "platform invisibility". By that I mean, that it is becoming invisible to the PC user where particular peripheral, data, and program resources reside. And it only makes sense - why should Fred in accounting have to be aware of exactly which machine contains his data.

A typical set of user instructions today includes:

(1) Choose the "HP" from your PC menu screen, (2) log on as :HELLO FRED,USER.FINANCE (press return), (3) :RUN FIN301.PUB.FINANCE (press return), (4) Press f1:reports, (5) type "101" into the report number field and press the grey "+" key, etc.

A "platform invisible" interface for our Fred would have him in the same familiar PC environment for all of his functions. To Fred, the HP-based finance system is as accessible as his spreadsheet system.

HP's NewWave is an example of such an integrated environment. It is also, however, so far in the "real soon now" category.

At some point, this invisibility may extend to the system developer. The specifications for a program in a platform-invisible environment need not include instructions as to where specific data files or programs reside. New distributed operating and/or application data base systems would be responsible for their physical locations, and, in fact, their locations could be adjusted "on the fly" based upon data traffic analyses, platform availability and capability, etc.

Dreaming? Perhaps. Perhaps not. Stay tuned.

* * *

Peripheral Sharing

Products in this category provide access from programs residing on one machine to peripheral devices residing on the other. They can spread the cost of expensive peripheral devices among many users, and provide access to peripherals which would otherwise be unavailable.

Some products in this category also provide network file services. These products turn the HP3000 into a PC file server.

Office Extend

Allows PC programs to print to devices residing on the HP3000 host. In addition, HP programs can spool output to PC printers. Access to HP3000 disk drives and network file services are provided by its HostDisk facility. PC's can also back up hard disk data to an HP3000 tape drive.

RSVP

Allows PC programs to print to devices residing on the HP3000 host. Utilizing Reflection, print files are spooled to disk and then transferred to the HP3000 for printing.

AdvancePrint

Allows PC programs to print to devices residing on the HP3000 host.

Resource Sharing

Allows PC programs to print to devices residing on the HP3000 host, utilize HP3000 disk space for PC file sharing, and back up data to an HP3000 tape drive.

Minisoft 2392

Part of this package allows the user to send printed output from the PC to host spooled printers.

Data Extraction / Download

Products in this category extract data from various sources on the host HP3000 and convert the information into one or more of several different PC application file formats.

The most common format is Lotus 1-2-3 file format, although several packages support other PC formats such as Excel, DIF, and comma delimited ASCII.

Other packages utilize this functionality as part of their operation - converting various HP3000 file formats into proprietary formats for use on the PC.

Most use auxiliary programs to perform the actual data download such as Reflection or AdvanceLink, although some contain datacomm facilities themselves.

DataExpress

Provides PC users with access to their data on the HP3000 for downloading. Accesses IMAGE, KSAM, and SD files. Produces formats readable by various PC application programs.

SpeedDoc

A complete word processing and office automation package. Can access IMAGE, KSAM, and MPE files to include data in word processing documents.

SheetMate

Translates HP3000 spooled output into Lotus 1-2-3 worksheets.

GUS/3000

A report writer which includes the capability of downloading output to the PC in Lotus 1-2-3 format.

DeskDirect

Adds commands to HPDesk which, when run in conjunction with Reflection, allows the user to send and receive PC files via HPDesk.

Visimage

A report writer and data extraction utility. Output can be downloaded and imported into various PC application programs.

Omniview

Operating as a Lotus Add-in, allows users to select, summarize, and download data into Lotus 1-2-3 spreadsheets.

Cooperative Computing

For the purposes of this paper, Cooperative Computing, also known as Distributed Processing, Cooperative Processing, Process Linking, or simply PC Integration, describes a computer system in which computing chores are shared by both the PC and the HP3000.

Once software designers iron out the inevitable problems with data/file type inconsistencies, and come up with an elegant "metaphor" for program to program communication, the linkage of the power of the PC and the HP3000 has enormous potential. In many ways, the two machines are complementary: the PC for data presentation and local manipulation, the HP 3000 for central data storage and transaction processing.

Software manufacturers recognized this potential soon after the PC became available in the HP3000 world - and the pace of product introduction has continued to increase ever since. The products presented here vary significantly in the degree to which they fit in the category of Cooperative Computing. Some depend upon it as an integral part of their processing. Some implement it for use in other products. Some offer it as an additional feature. All of them, however, help us to see the potential of this powerful idea.

Cooperative Services

Allows PC programs to call MPE Intrinsics and user-written SL routines directly.

KOS

KOS really belongs in its own category. This product emulates an HP3000 on a PC. I would have to say that this takes the concept of PC/HP3000 integration to the point of a Vulcan mind-meld. However, disregarding its several interesting aspects, it also allows programs running in HP3000 emulation mode on the PC to access files and data bases on a "real" HP3000 and so has been placed in this category.

VPLUS/Windows

Provides access to VPLUS/3000 programs from within the Windows environment on the PC.

Office Extend

One of the capabilities of its HostDisk facility includes the ability to open an MPE program as a "file" from a PC based application program. Program communication then takes place utilizing STDIN/STDLIST as "pipes".

Synergist

Distributes application processing to the PC and uses the HP3000 for storing and retrieving data from a database management system.

Speedware

A fourth-generational development and reporting system providing a common application development environment for both the PC and HP3000. Both application code and data storage can be distributed as needed between the HP3000 and the PC.

More Cooperative Computing

Powerhouse

A fourth-generational development and reporting system providing a common application development environment for both the PC and HP3000. Both application code and data storage can be distributed as needed between the HP3000 and the PC.

Dispatch

Provides the capability to compose and schedule mail to HPDesk running on the host HP3000. Mail can then be received and/or transmitted during a connection with the host.

NewWave

A Windows/OS2-based environment, this product is forming the backbone of HP's future integration efforts. Currently, rudimentary integration with other HP products like Information Access is available.

PPL

Provides a data path from PC applications to HP3000 applications. Used for custom PC/HP3000 systems both user-written and those produced by some third-party vendors.

AdvanceMail

Provides the capability to compose and schedule mail to HPDesk running on the host HP3000. Mail can then be received and/or transmitted during a connection with the host.

Oracle

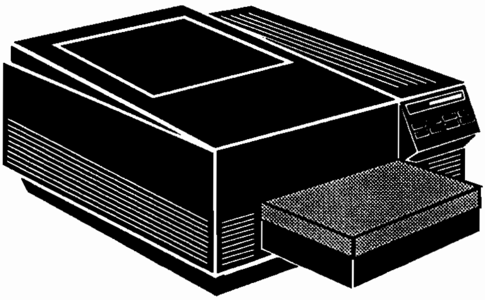
A relational database system, Oracle also has the capability of distributing a single database across PC's, minis, and mainframes.

Vendors

AdvanceMail	Hewlett-Packard
AdvancePrint	Hewlett-Packard
Cooperative Services	Hewlett-Packard
DataExpress	IMACS Systems Corporation 2825 Eastlake Avenue East, Suite 107 Seattle, WA 98102 (206) 322-7700
DeskDirect	WalkerRicher&Quinn 2815 EastLake Avenue East Seattle, WA 98102 (206) 324-0350
Dispatch	WalkerRicher&Quinn 2815 EastLake Avenue East Seattle, WA 98102 (206) 324-0350
GUS/3000	Comprehensive Systems, Inc. 720 King Georges Road, Fords, NJ 08863 (201) 225-9670
KOS	Kydor Computer Systems 1707 E Collins Blvd. Richardson, TX 75081 (214) 480-0170
Minisoft 2392	16315 NE 87th Suite B101 Redmond, WA 98052 (206) 883-1353
NewWave	Hewlett-Packard
Office Extend	Fransen/King, Ltd. 16400 Southcenter Parkway Seattle, WA 98188 (206) 575-1570
Omniview	Dynamic Information Systems Corporation 910 15th Street, Suite 640 Denver, CO 80202 (303) 893-0335
Oracle	Oracle Corporation 20 Davis Drive Belmont CA 94002 (800) 345-DBMS
Powerhouse	Cognos Incorporated 3755 Riverside Drive Ottawa, Ontario (508) 535-7350
PPL	WalkerRicher&Quinn 2815 EastLake Avenue East Seattle, WA 98102 (206) 324-0350
Resource Sharing	Hewlett-Packard Company
RSVP	WalkerRicher&Quinn 2815 EastLake Avenue East Seattle, WA 98102 (206) 324-0350
SheetMate	Hillary Software, Inc. 309 Morris Avenue, Spring Lake, NJ 07762 (201) 974-8484
SpeedDoc	Bradford Business Systems, Inc. 25301 Cabot Road, Suite 201, Laguna Hills, CA 92653 (714) 859-4428
Speedware	Infocentre Corp. 7420 Airport RD., Suite 201 Mississauga, Ontario L4T 4E5 (416) 678-1841
Synergist	Gateway Systems Corporation 2400 Science Pkwy Okemos, MI 48864 (517) 349-1520
Visimage	VitalSoft 1916 Old Middlefield Way Mt. View, CA 94043 (415) 965-4494
VPLUS/Windows	Hewlett-Packard

Understanding Printer Sharing Solutions

by Thomas Durr
VP Marketing
Wilco-AMS Inc.



Wilco-AMS Inc.
473 Macara Avenue, Suite 702
Sunnyvale, California 94086
408/746 3993

4129-0

Printer Sharing Device Definition:

Printer sharing systems make it possible for two or more people to access and share the cost and performance of workgroup printer resources. Sharing significantly lowers the cost per user, making it easier for companies to justify the purchase of high resolution printers, like Hewlett-Packard LaserJet printers, plotters and color ink jets. Most printer sharing devices include buffering to reduce the normal wait associated with printing as well as the wait when two or more users print simultaneously.

Printer Sharing Device Criteria:

Share printer and printer accessory cost.
Give more people access to the printer or printers.
Buffer print jobs to increase user productivity.
Run the printer at top speed over long cables.

Printer sharing devices can be surprisingly practical products. When properly implemented, they provide significant equipment savings and increased user productivity, however, printer sharing system performance and reliability varies widely. To *buffer incoming print jobs and simultaneously run the printer at top speed*, a printer sharing system must be able to transfer data at very high data rates. The measure of a printer sharing system's ability to transfer data is called *throughput*. Many printer sharing systems offer throughput far below the processing speed of the LaserJet III. These underpowered printer sharing systems can seriously degrade LaserJet printing performance, making users wait needlessly.

Part One of this presentation defines the characteristics of *state-of-the-art* printer sharing systems. Part Two focuses on the various ways printer sharing systems can be used to share printers as stand-alone solutions or working in conjunction with LANs and minicomputers.

The state-of-the-art printer sharing system should come as close as possible to providing each user with the performance of a dedicated printer, or better yet, a dedicated printer with a high speed buffer.

Part One

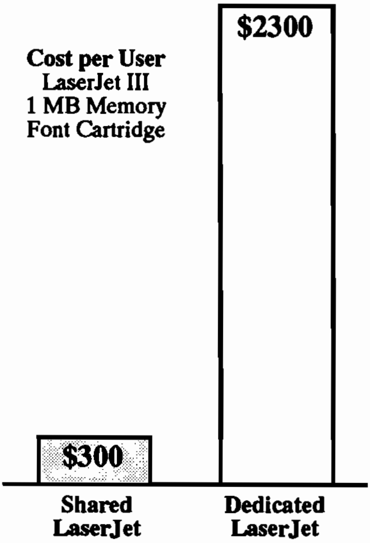
Printer Sharing Nuts and Bolts

Printer Sharing Economics:

The HP LaserJet III costs about $1600. Adding a megabyte of memory costs around $250 and a font cartridge costs around $450. The cost of giving *each user* a LaserJet III with memory and a font cartridge is approximately $2300. A ten computer port printer sharing system can bring the cost per user for a LaserJet III down to $300 per user! The cost of providing 100 people with 100 dedicated LaserJet IIIs is $230,000 compared to $30,000 using ten port sharing devices.

Printer Sharing Device Savings:

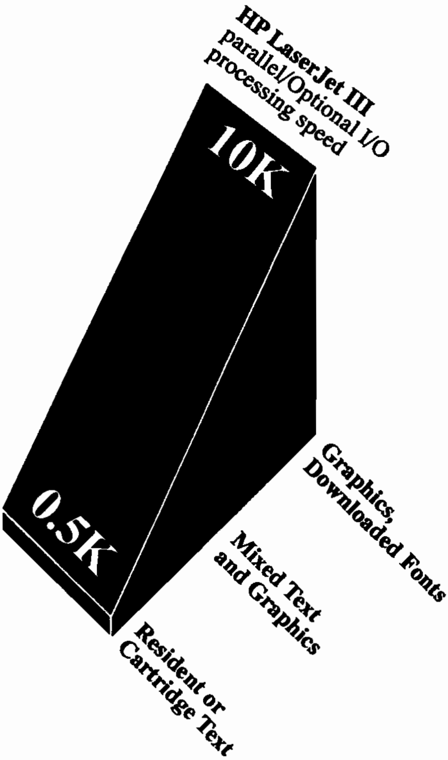
HP LaserJet III with accessories	**$2300**
Average 10 Port Sharing System	**+ 700**
	$3000
	÷ 10 users
Save $2000 per user!	**$300 per user**



Hewlett-Packard LaserJet Printing Speed:

Printing is one of the slowest computing processes. Even a high powered 386 Vectra can spend a number of *hours* per week *locked-up*, waiting for the printer. The advent of graphics printing and font downloading has increased the size of an average page from just a few thousand characters or bytes to as many as a million characters or bytes per page. These relatively huge jobs amplify printing lock-up. Fortunately, the LaserJet can process graphic and font data at very high speeds through its parallel port or through its Optional I/O slot as shown in the *LaserJet Performance Wedge* illustration at right. Test measurements show that the LaserJet can receive pure graphics and fonts data at better than 10,000 characters per second through its parallel and Optional I/O ports. Most files include a mixture of text and graphics data which the LaserJet processes nominally at approximately 6,000 characters per second through its high speed ports. Conversely, transmission to the LaserJet's serial port is typically limited to 9600 baud *(less than 1,000 cps)*.

Increasing the LaserJet's internal memory does not improve the speed at which the printer can take in print jobs. It simply makes it possible to print a more complex page or store more downloaded fonts.



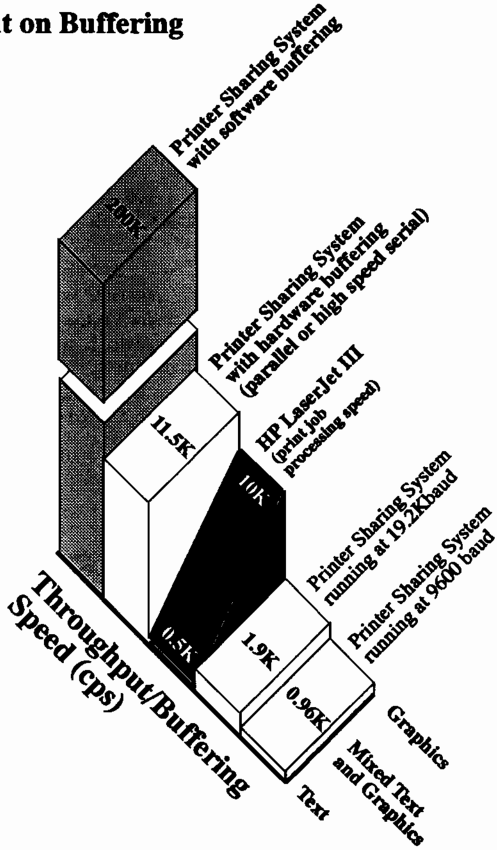
To print graphics and download fonts to a LaserJet at top speed, the printer sharing system needs to be attached to either the LaserJet's parallel port or its Optional I/O slot.

The Impact of Throughput on Buffering and Printing Speeds:

While the LaserJet can process data at up to 10,000 characters per second, typical printer sharing systems, including LANs, limit total data throughput to less than 3,000 characters per second (cps). Single inputs are limited to less than 1,000 cps if only 9600 baud is supported. Even parallel interface sharing systems may not have the throughput to run at high speed parallel data rates. Systems that offer high serial baud rates may not be able to sustain those baud rates continuously due to throughput limitation. A one megabyte page (the equivalent of one million characters) containing graphics and fonts that takes just over three minutes to print through the LaserJet's parallel port can take *sixteen minutes* to print through a 1,000 cps sharing system. *Both the computer and the printer are locked up for sixteen minutes.* Adding more buffering doesn't help since the problem is buffering speed, not buffering capacity.

To run the printer at top speed, the printer sharing system must be, at minimum, as fast as the printer's parallel port. To provide the benefits of buffering and run the printer at top speed, the printer sharing system must be *a lot faster* than the printer. State-of-the-art systems offer throughputs in excess of 20,000 cps and serial input rates up to 115.2 Kbaud (11.5K cps).

These relationships can be illustrated by placing the *LaserJet Performance Wedge* next to the Printer Sharing Systems' buffering/throughput measurements as shown above right:

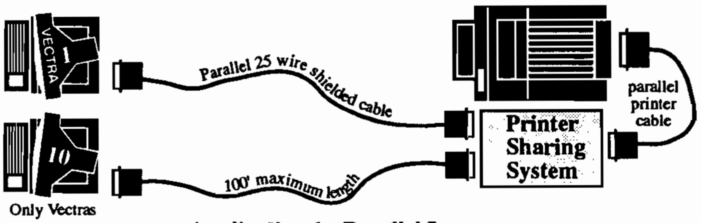


To provide the benefits of buffering and run the printer at top speed, the printer sharing system needs to be a lot faster than the printer.

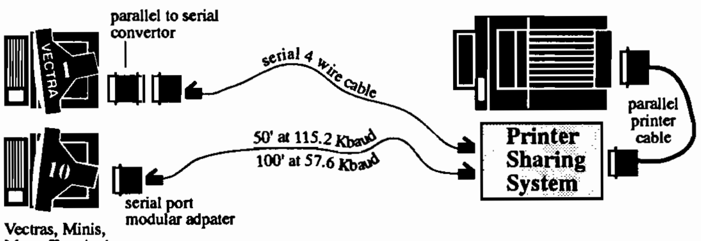
High Speed Printer Sharing - Serial versus Parallel Inputs

We know that a state-of-the-art sharing system needs to be attached to the LaserJet's parallel or Optional I/O slot. What's the best way to connect computers to the sharing system? MS-DOS limits an HP Vectra from transmitting any faster than 9600 baud through its serial port even though the serial (COM) ports support data rates up to 115,200 baud. The parallel interface is very fast, however, cables are typically limited to fifteen feet. A few sharing systems offer extended parallel receiving capability that allow parallel cables up to 100 feet long. These cables are bulky and relatively expensive when compared to serial cable schemes.

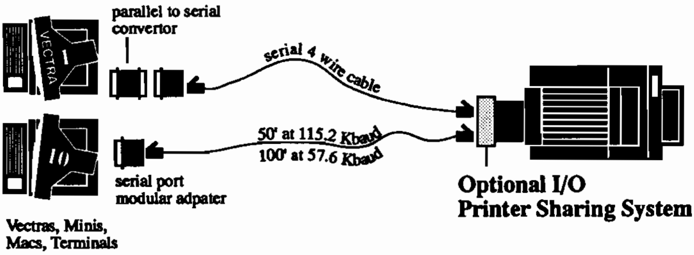
Providing automatic *serial to parallel conversion*, new printer sharing systems offer high speed serial inputs and high speed parallel or Optional I/O slot connection to the LaserJet. These systems include high speed serial software drivers for Vectras that replace the MS-DOS serial port commands. Connection to the printer sharing device is made through low-cost, telephone style, 4 wire cables. Connection from the printer sharing device to the LaserJet is made either through the LaserJet's high speed parallel port or through its Optional I/O slot. If the Vectra is lacking a spare serial COM port, a *parallel to serial convertor* can be used at the Vectra side.



Application 1: Parallel Inputs



Application 2: High Speed Serial Inputs



Application 3: High Speed Serial Inputs

All three of these applications represent high speed printer sharing system examples. Applications two and three utilize high speed serial to parallel conversion. Assuming that each of the ten computers needs to be located an average of fifty feet from the sharing device, then the total cable distance requirement will be 500 feet. Parallel cable costs approximately $1.50 per foot while serial telephone cables cost approximately $0.36 per foot. Cable costs with the parallel interface printer sharing system would be $750 and cable costs for the serial telephone cables would be $180.

Serial input systems are usually the way to go:

1. **75% less expensive cable than parallel.**
2. **Vectras, HP 3000s, terminals and Macs can be connected serially.**
3. **Computers can be connected hundreds of feet away.**

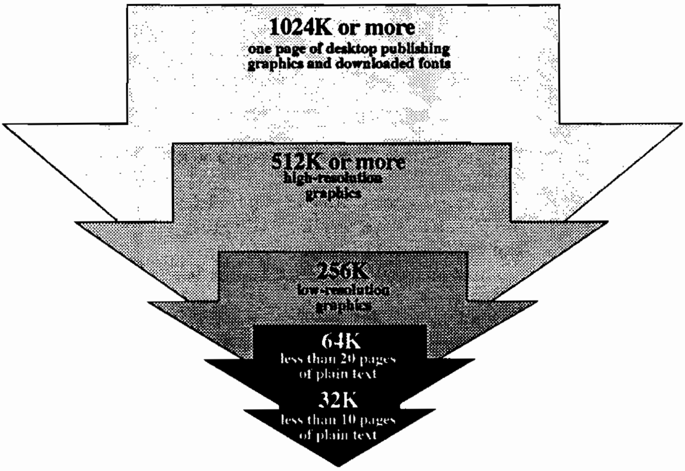
Serial Baud Rates and Cable Lengths

Serial cable lengths are directly influenced by serial baud rates. The following is a conservative estimate of cable lengths at various serial baud rates. Twice the length at each baud rate may be possible.

Baud Rate	Cable Length (feet)
9600	500'
19200	300'
38400	200'
57600	100'
115200	50'

Print Job Size and Buffering Capacity

A single page of text using resident or cartridge fonts is only about 3 kilobytes of data. A singlepage containing graphics is usually more than 512 kilobytes of data (512,000 characters). Pages as large as 1000 kilobytes (one million characters) are common in desktop publishing applications such as Ventura or PageMaker. A single page of DTP output is *300 times larger* than a page of text. Hardware buffer capacity in many printer sharing systems is limited to 256 kilobytes; not enough for memory hungry graphics and DTP print jobs. Larger buffers can be ordered but they can be expensive and many are too slow. In fact, a buffer with a throughput of less than 1,000 characters per second is pretty much a complete waste of money for graphics printing since the buffer's input data rate is far less than the speed of the LaserJet. Fortunately, the buffering in new state-of-the-art printer sharing systems is very fast - making practical, the purchase of larger buffers.



A single page of graphics and downloaded fonts is 300 times larger than a page of text. To minimize printing lock-up, use a high speed printer sharing system with enough buffering capacity to handle the workgroup's printing job size and usage.

Buffering for Text, Graphics and DTP Printing Workgroups

Many *low throughput* printer sharing systems include small buffers ranging from 64 to 256 kilobytes. These systems and buffers are adequate for workgroups printing small text print jobs. 256 kilobytes is enough to hold eighty pages of plain text data.

Workgroups printing simple graphics like lines, boxes and circles should purchase high throughput sharing systems with more than 512 kilobytes of buffering.

An HP Vectra workgroup doing desktop publishing requires a high throughput printer sharing system combined with a high powered buffering solution. In lieu of purchasing a video laser controller card for your computer and dedicating a printer, ***high speed buffering software*** can provide the kind of buffering capacity and speed needed for DTP printing. High speed buffering software lets each user set up a print buffer specifically designed for their particular printing requirements. Buffering capacity is only limited by the availability of hard disk, EMS or Extended memory. When a hard disk buffer is used, buffering capacity is only limited by the available hard disk space. Buffering speed to hard disk can achieve 50,000 characters per second. Using high speed buffering software, a one megabyte file would be buffered in less than thirty seconds. Extended or EMS buffering is even faster. Of course, buffering speed depends on how fast the computer or the application software can generate the data. Vectra 286 machines will do pretty well. High speed buffering software on a 386 machine provides relatively huge time savings.

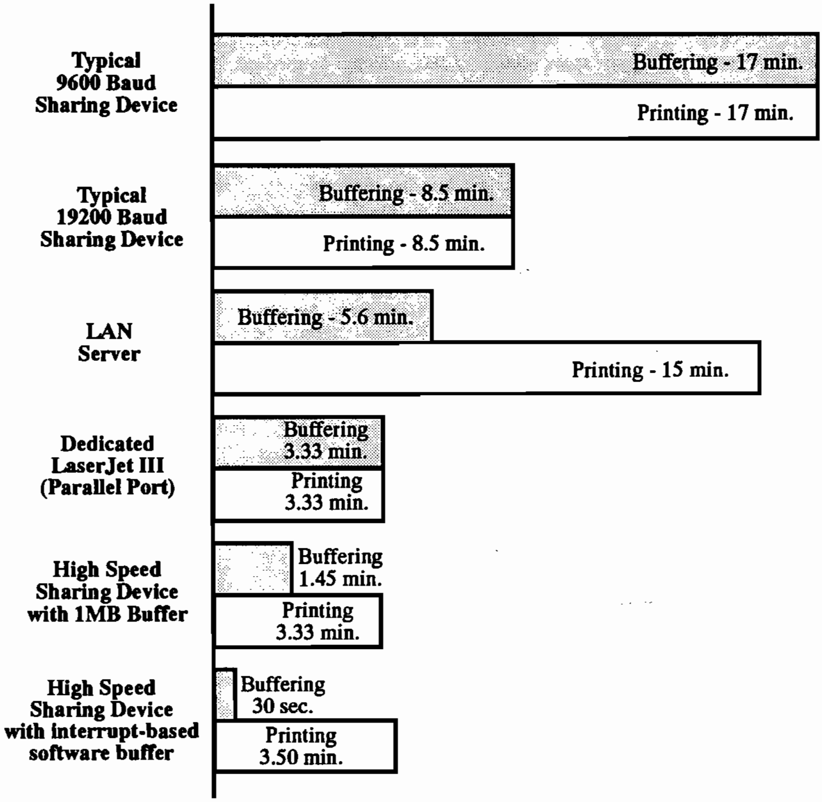
Like printer sharing throughput, buffering software products are *not created equal.* Really fast buffering software products use the Vectra's DOS interrupt to control the buffer input/output process. The Vectra will be *interrupted* by the printer and transmit only when the printer is ready to receive data. This method also uses the least amount of foreground processing time on the Vectra. Other software buffering solutions, including LAN spoolers, use a "time-slice" method. With a time-slice buffer control, a fixed percentage of the processor's clock is used for buffering. Time-slice buffer control is always much slower than interrupt-based software buffering.

Workgroup Printing	Buffering Capacity
Text	256 kilobytes
Simple Graphics	1 megabyte
Desktop Publishing	Interrupt-based software buffering

Buffering and Printing Speed Benchmarks:

Testing clearly illustrates the performance degradation caused by slow printer sharing solutions as well as the advantages of high speed buffering. These tests show the wait for buffering and printing a single complex page. Imagine the printing lock-up and productivity loss that occurs when multiple users are sending complex pages!

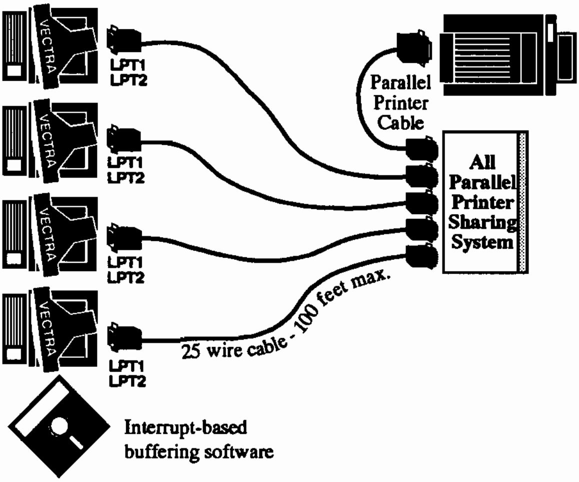
Tests are based on transmitting a one megabyte file to an HP LaserJet III.



Part Two

Printer Sharing Applications

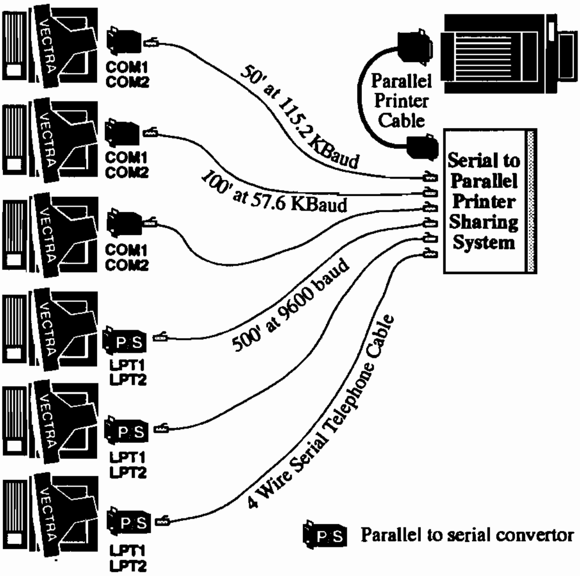
HP Vectras and Compatibles -> LaserJet:



Parallel to Parallel sharing systems are ideal for HP Vectra workgroups located *near the printer.* Interrupt-based buffering software can be used for desktop publishing environments.

Pros: LPT# port can be used rather than a serial (COM) port or a parallel/ serial convertor.

Cons: Expensive, bulky cables and limited cable distances.

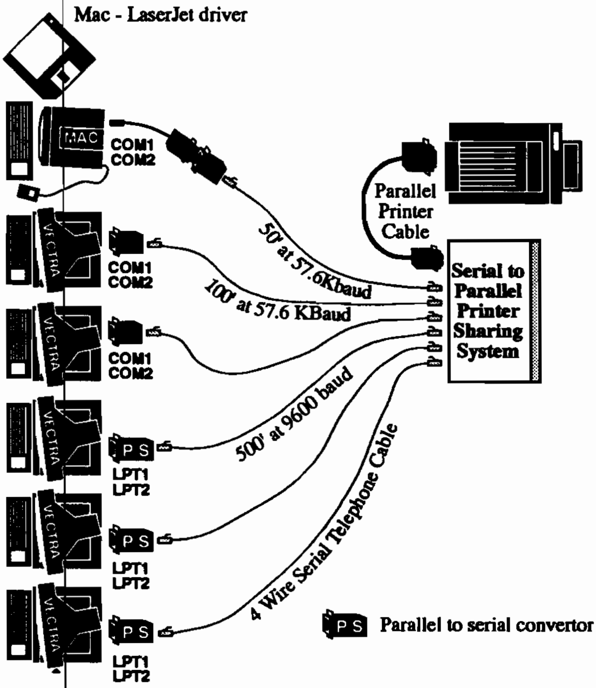


Serial to Parallel sharing systems are the best solution for spread out workgroups. They can also be located in a remote wiring closet. Interrupt-based buffering software can be used for desktop publishing environments.

Pros: Low cost cables.

Cons: Requires a serial (COM) port or a Parallel/ Serial convertor.

HP Vectras and Macs -> LaserJet:

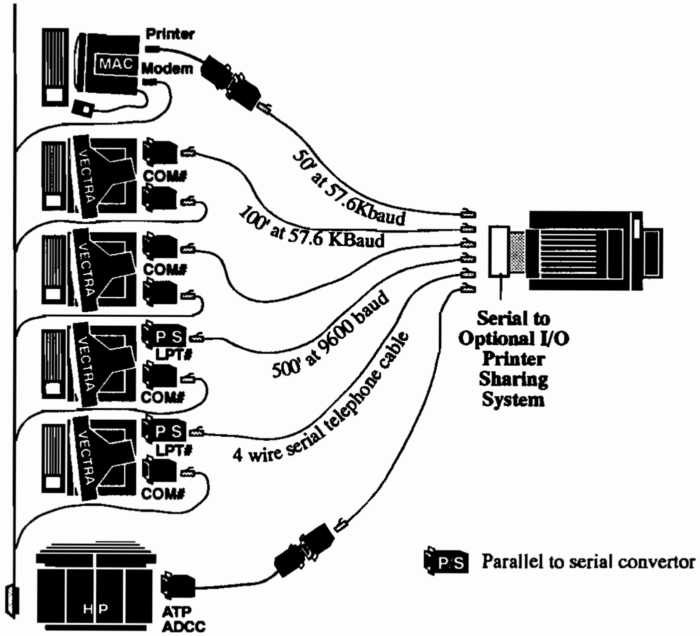


Serial to parallel printer sharing devices can be used to simultaneously share a LaserJet between HP Vectras and Apple Macintoshes. A special LaserJet driver(Chooser) must be purchased for the Mac. Usually, these drivers can be configured to take advantage of the printer sharing system's faster serial baud rates (57.6 KBaud). Conversely, the LaserJet's own serial port is limited to 19.2 Kbaud.

Pros: Macs can be connected to a LaserJet a lot easier than connecting HP Vectras to AppleTALK.

Cons: Mac drivers can be slow although the faster serial baud rate and buffering provided by the printer sharing system can help significantly.

HP Vectras, Macs, HP3000 -> LaserJet:

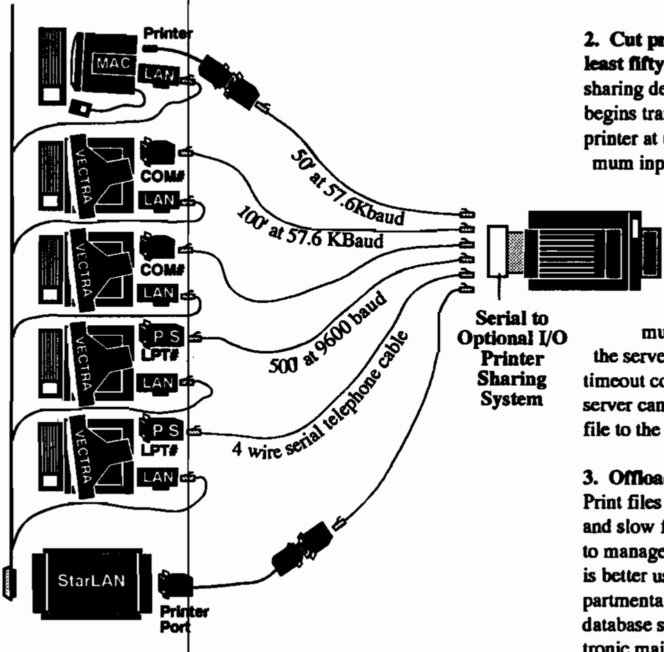


Serial to parallel printer sharing devices can be used to simultaneously share a LaserJet between personal computers, terminals and the HP 3000. Some sharing systems include special support for HP 3000 terminal type 22 or 26 status checking. The serial port on the HP LaserJet series II supports terminal type 22/26 status checking. Newer LaserJets (IID and III) do not.

Pros: Personal computer users can print through the HP 3000 spooler or directly. Users printing through the HP 3000 spooler receive printer status (paper out, jam...) when Terminal Type 22/26 status checking is supported.

Cons: HP Vectra connections require a COM port or a Parallel/Serial convertor.

HP Vectras, Macs, LAN Server -> LaserJet:



Serial to Parallel printer sharing devices can be used to simultaneously share a LaserJet between personal computers, terminals and a LAN server. Users can print through the LAN or directly.

Pros: Four major benefits.

1. Print directly or through the network server. The ability to print directly is most important for applications that generate graphics and download fonts. In many cases, the LAN's memory resident module uses too much of the Vectra's conventional memory making it difficult to run graphical applications.

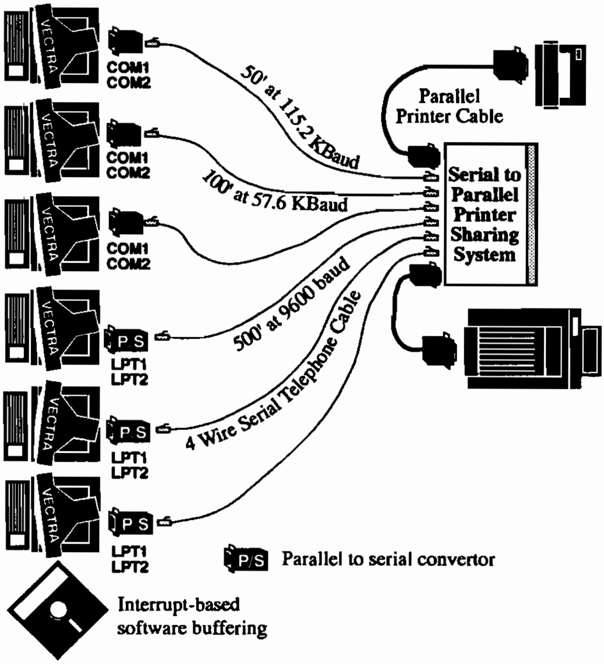
2. Cut printing time by at least fifty percent. A printer sharing device immediately begins transmitting to the printer at the printer's optimum input rate. Printing through a LAN requires a number of steps that slow the printing process - The entire print job must be received by the server and an end-of-file timeout completed before the server can begin sending the file to the printer.

3. Offload the LAN server. Print files are relatively large and slow files for the server to manage. Server throughput is better used to handle departmental file transfer, database sharing and electronic mail requirements.

4. Achieve 200,000 cps buffering speed on each HP Vectra. Vectras directly printing complex graphics and downloading fonts can utilize an interrupt-based software buffer for super fast print spooling. Conversely, most LAN software spools at 3-5,000 cps.

Cons: Requires additional cabling.

Sharing Multiple Printers

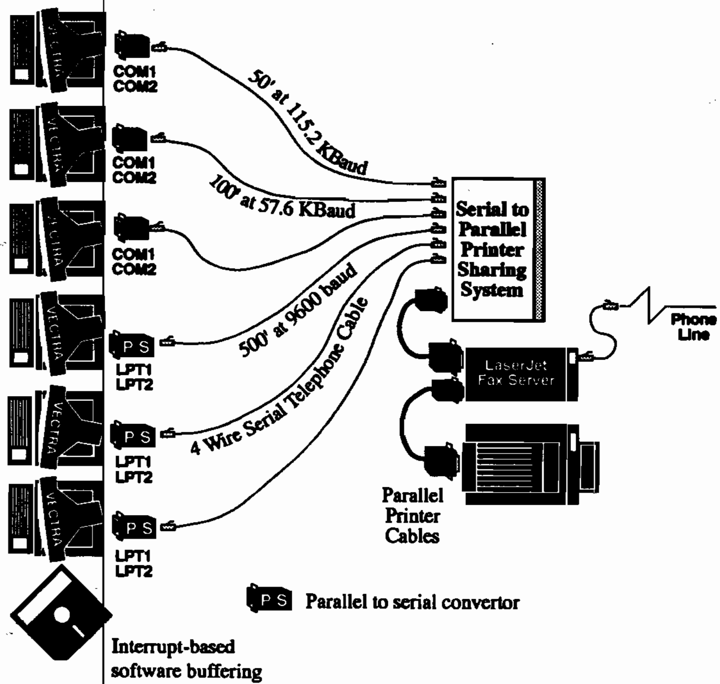


Serial to Parallel sharing systems can be used to share multiple printers. Each user can connect to either printer and both printers should be able to print simultaneously. Some systems provide the ability to print on the *first available* printer. Switching between printers is accomplished by preceeding a print job with a printer selection code.

Pros: Each user can access different types of printers.

Cons: Users need to configure their software according to the type of printer and, at the same time, they need to keep track of which printer they have selected by sending the printer selection code.

Sharing Printers and Fax Servers

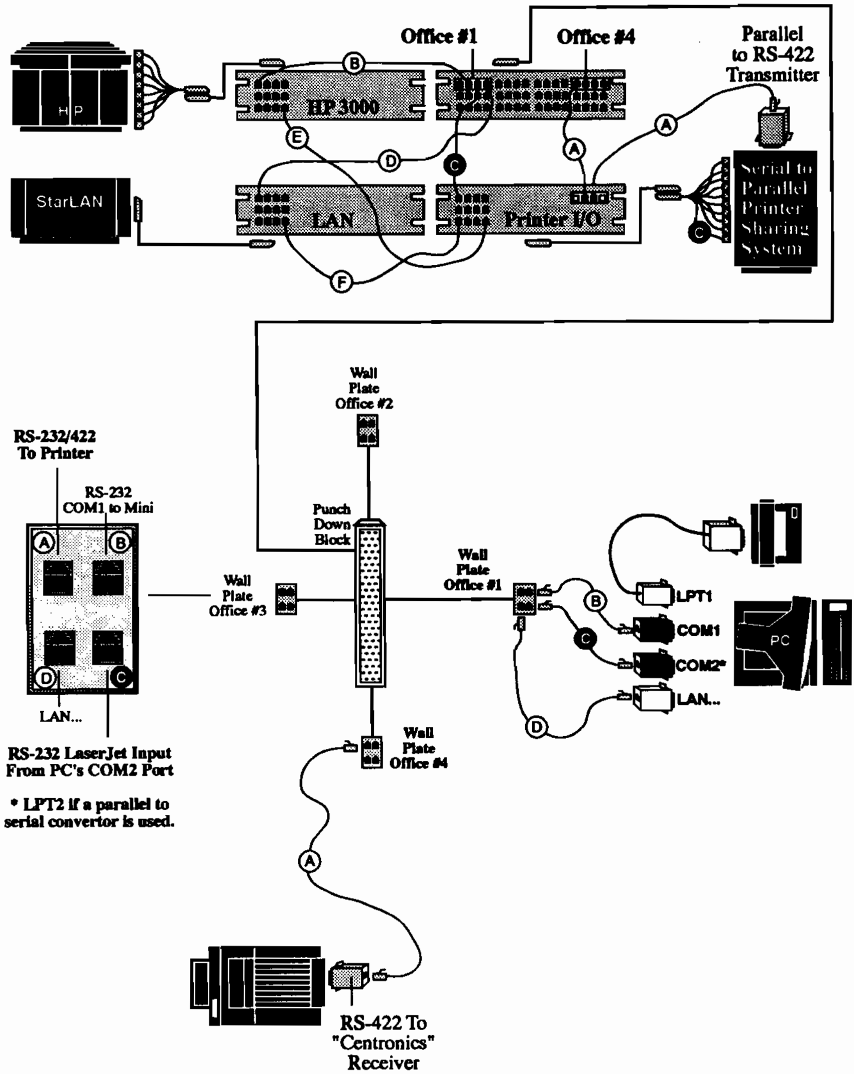


Desktop Serial to parallel printer sharing systems are ideally suited for sharing an integrated laser printer and laser printer fax server as shown above. Like graphics print jobs, files converted to fax format are relatively large files. High throughput is necessary for effic ient operation.

Pros: Significant equipment savings. Each user can print or transmit a fax. Incoming faxes are printed on regular paper.

Cons: No way to view incoming faxes before printing.

Wiring Closet Printer Sharing Application



Summary

Summary

For printing graphics and downloading fonts to a LaserJet *at top speed*, the printer sharing system needs to be attached to either the LaserJet's parallel port or its Optional I/O slot.
The LaserJet's parallel and Optional I/O ports are at least five times faster than its serial port.

To provide the benefits of buffering *and* run the printer at top speed, the printer sharing system needs to be faster than the printer.
If the printer can receive data nominally at 6,000 cps, than the printer sharing system needs to be able to receive data into its buffer at better than 6,000 cps. High throughput hardware solutions can receive data at roughly double the 6,000 cps rate which cuts the printing lock-up in half. Interrupt-based software buffering provides the fastest buffering available for HP Vectras and compatibles.

Systems that utilize high speed serial inputs and serial to parallel (and Optional I/O) conversiou are usually the best way to go.
A printer sharing system with serial inputs lets you connect all types of computers, not just HP Vectras. Computers can be located hundreds of feet away and cables are seventy-five percent less expensive.

Once you have determined that the printer sharing system has adequate throughput, make sure you purchase enough buffering capacity.
When purchasing a printer sharing system, buy an adequate amount of buffering depending on your workgroup's printing requirements. Just one user in the workgroup that transmits DTP print jobs (1000+K) can monopolize the sharing device's buffer and the printer. If the buffer is too small, other computers that attempt to print may be locked up for extended periods of time waiting for the first user's job to finish.

If you hate to wait, look for sharing solutions that offer a full range of buffer capacity.
A typical printer sharing device with a 256K buffer is sufficient for text and low resolution print jobs. High speed printer sharing systems with 1MB hardware buffers are needed for high resolution and occasional desktop publishing print jobs. With virtually *unlimited* buffer capacity, interrupt-based software buffers are the only way to go for heavy desktop publishing workgroups.

Beware of underpowered printer sharing systems. When evaluating a system, be sure to do some timing tests.
Many printer sharing devices save money in equipment costs but end up costing much more in lost man hours due to inadequate throughput. A sharing device that offers high speed serial or parallel inputs may not have the system throughput to sustain fast data rates. High speed, high throughput printer sharing devices equipped with sufficient buffering capacity really can save a lot of money by saving both equipment cost and time.

4131

Using Information Access to Get SQL Data to PC Users

Author

Robert Ross
Hewlett-Packard
Data Management Systems Division
Roseville, California
National INTEREX Conference: August, 1990

Making Decisions

How Do You Make Decisions?

Decisions are a daily part of life. People spend varying amounts of time pondering over decisions for different reasons, often depending on how important the decisions are, and whether they relate to personal or business matters. What to have for breakfast (how many types of cereal do you have?) may be trivial, but daily business decisions can be extremely complex and depend on many different variables.

To choose between similar alternatives based on these variables may take experience, intuition, or even guesswork, but the most common basis for choice will be hard data. This data may come in the form of orders, shipments, personnel records, forecasts or even an opinion poll, but to be valuable to the person making the decision, it must be presented in an easily understood format.

What Is Today's Data Environment?

Most medium to larger sized companies today use mini or mainframe computers to keep track of the majority of their accounting and manufacturing data. This approach lends itself to centralized computing facilities where control over sensitive data can be more easily managed.

This data should be stored in some sort of database system. More and more of these database systems today allow data to be retrieved using SQL (Structured Query Language) commands that are fairly portable across multiple systems. This commonality within the SQL language significantly reduces the learning curve for users and programmers.

How Do You Get Data to Make Decisions?

As the business environment becomes more and more competitive, not only do managers need the right data in an easily understood format, but they also need it before its usefulness expires. Getting the data you need today at the end of the week doesn't do you much good. Making the customer wait for several minutes on the phone while you try to find their records will eventually produce nothing but frustration and ex-customers.

There are several ways to get this data. You could:

- Pay a consultant to get it for you (and pay again next time ...)
- Go through the learning curve yourself so that you are able to retrieve the data when you need it
- Train some of your personnel to be able to get the data when asked
- Get some integrated software do accomplish the task for you, with minimal learning on your part

It is this last choice that I will be covering in this paper.

Can We Make Getting Data Easier?

Assuming we have the data in a central, accessible place using modern database technology, we also have an end-user language called SQL that allows quick ad-hoc queries and reports without having to place requests with the programming staff. So why aren't more end-users taking advantage of this and looking up data and doing reports themselves?

Even with the sophisticated and widely used tools available today, many users are put off by the learning curve associated with implementing new ways of doing things, though it will eventually mean getting their job done more quickly. The tools are there, the data is there, but users still have trouble getting access to it.

Concentrate on the Tasks Performed Most Often

Unfortunately, there is no such thing as a completely intuitive user interface that still allows the power to accomplish complex tasks. But we **can** and **should** separate out the tasks performed most often and concentrate on making them the easiest to do.

There are more people in most organizations today trained in some way to use the personal computer than any other aspect of the computerized data life cycle. It therefore makes sense to combine the need for quick and easy data retrieval with the large training investment in PCs to provide the answer to our problem. The rest of this paper uses a case study as an example of how PC software can be combined with large central database systems to give the end user a quick path to viewing the data essential for them to make decisions and keep customers happy.

A Case Study - How was the Problem Defined?

This example is based on a solution offered to a large manufacturing customer who had a particular set of needs for a particular set of users. These users were fairly inexperienced PC users - mostly customer service representatives who needed to be able to find the status of a customer order (which was equivalent to the manufacturing lot number) while the customer was waiting on the phone for a progress report, or a few product line managers who were tracking their work-in-process as their parts were being manufactured for use in a sub-assembly.

How Was Their Data Stored?

The customer had recently migrated their manufacturing environment to ALLBASE/SQL on an HP 9000 Series 850. To simplify the problem, we cut a narrow slice out of their total data life cycle and concentrated on the needs of the users previously mentioned. They wanted to look at particular part number given to them by the customer - but they also needed to view a particular lot number within the part number because each lot belonged to a certain customer.

The actual data was stored in two separate tables - a **Part Table** and a **Lot Table**.

The **Part Table** was a master description table where each row contained columns that described all aspects of each part that was manufactured in the plant - the part number, description, price, product manager, run time, material cost, etc.

The **Lot Table** is based on the **Part Table** in the sense that each lot pertains to only one part number. However, a part number may have 1 or more lots with each lot being differentiated by a customer order. For example, if 4 customers order part number 1010-2002-34 in quantities of 100, 1200, 20 and 4000, the **Part Table** will still contain only 1 row for the part number, but the **Lot Table** will have 4 rows, one for each of these orders.

How Did Users Want to See the Data?

The users wanted to see a combined view of the these two tables, with information from both of them, but leaving out many of the columns from both tables. They were getting a printed listing of all ordered lots for each part number. These listings would combine the information from both tables in a way that the representatives were accustomed to, and they wanted to continue viewing it in the same way, although modified somewhat to fit on the terminal display, such as the following, which shows a single lot number:

```
Part Number:  1010-2002-34            Customer No. : ABC-1010
Description:  2"center-pin elbow      Company Name : Aston Britton Const.
Lot Number.:  16-8480A                G/L Income Acct  3010 - 29  A/R Acct  1100
Contract Due Date:       06/22/93     Report % Compl'd   45.00%  on  04/01/93
Contract Amount..:      35,700.00     Orig/Rev Budget
Contract Number..:         61002C     Est/Budget/JTD Profit % 0.00/ 22.00/355.00
Job Type [FF/TM/CP/NB]         FF     Rev. Contract Amt.         35,700.00
Customer's Retention %       0.00     Rev. Due Date          / /
Contingency Fee %            0.00     Last Revision Date     / /
Cost Method [PC/CC]            CC     JTD Actual Cost            3,644.91
Job Manager                  JOHN     JTD Direct Cost            3,644.91
Job Status [A/I/C]              A     Billable Amount                0.00
Status Date              04/01/93     Amount Billed                  0.00
Est. Start Date          03/27/93     Amount Received                0.00
Act. Start Date          03/27/93     Over/Under Billed              0.00
Est. Close Date          04/05/93     Last Billing Date      / /
Remark:
```

These part and lot number printouts were generated over the weekend and would become more and more outdated by the end of the week. When customers would call in inquiring about the status of their order, they would often be given incorrect information since the customer service representative did not have time to find the correct shop floor, put in a call to the building and ask the floor supervisor about a certain order, and then wait while the information was being manually retrieved to finally pass back to the customer. On top of this, the floor supervisors refused to spend so much of their day looking up order status information for every customer that called.

ISQL Was Available, But Not Used

Once the customer had migrated over to the ALLBASE/SQL database system, it would have been fairly easy for the service representative to log on to the host system, run ISQL (Interactive SQL), execute a SELECT statement WHERE PART-NUMBER is equal to the one being asked about by the customer, and get a "quick and dirty" listing of all lots within that part number. However, this method was unsatisfactory for a couple of reasons.

First, most of the daily work performed by the customer service representatives was PC oriented, and the time to run the terminal emulator, log on to the host, and run ISQL was perceived as too long.

Second, the output from this method gave the information in a linear and columnar format and the representatives were unable to find the information easily enough as they were trying to scroll through several vertical and horizontal screens of information.

Third, most users were unable to grasp the concept of using the SELECT statement properly to join the two tables together that contained the part number and lot information. Even when shown how to do this operation, their perception after many attempts was that it was too much work to get the syntax right.

What Did Users Really Want to See?

A typical phone exchange between a customer and the service representative consisted of the customer giving a part number, a customer number, and a customer name for verification. The representative would then flip through the weekly work-in-process report to find the part number requested by the customer. Since all lots of the part number would be together, the representative could match up the customer number with one of the lots for that part number and quickly let the customer know what the estimated completion date would be.

On the new system, they wanted to ask for the same information and immediately type in the part number from the customer while they were asking for the additional information. They wanted to have a display within a few seconds that would show them the same information that they were accustomed to seeing on the weekly printouts.

They wanted to see a single lot on each screen, and also be able to flip backwards and forwards through the lots to find the correct one, since large customers would often have multiple orders, and therefore more than one lot per part number.

What Did We Do For a Solution?

The solution that combined ease of use, little training, and the display features asked for by the users turned out to be a four-step process:

- Execute an Information Access command file from the PAM menu on the PC screen
- Enter the part number that the customer was asking about
- Continue talking and ask the customer for more information while the results were printing
- Scan the output and find the right lot number on the printout to give the requested information back to the customer

What is Information Access?

Information Access is one of several products on the market today that can accomplish the tasks presented here. It is a combination of code on the PC and the host (either an HP 3000 or an HP 9000) that allows end users to work through a common PC interface to retrieve and report on data from various database systems. For Information Access, this data may be in the form of TurboIMAGE, KSAM, MPE file, or SQL data on MPE V, MPE XL, or HP-UX systems. Several PC database and spreadsheet formats are supported also, such as Lotus 1-2-3, Symphony, dBASE, Rbase and Condor.

What are Information Access Command Files?

The user interface of Information Access lends itself to interactive queries, reports, data reduction and summation. However, there is also a script or command language with commands and a process flow that is similar to what users see when executing an interactive session.

The command file may be generated using any text editor or word processing program that can save files in a simple ASCII format. Once the command file has been created, the Information Access program can be set up to automatically execute a given command file when run with an "INFO=xxx" parameter, which means that we were able to set up a command file to be run from a menu choice on the PAM screen.

How Did the Command File Work?

The command file is simply a line by line sequence of commands that tell Information Access to ask the user for a part number, log on to the host computer (if not already logged on), execute the SELECT command that would create and retrieve the view table containing all lots for the given part number, and finally output each row to a local printer using the built-in Report Writer before exiting. It would look similar to the following:

```
*
* Get a part number from the user
*
INPUT "Please enter part number:" inputpartnum
*
* Establish LAN access to remote ALLBASE/SQL tables on the HP 9000
*
SET_CONNECTTYPE "ALLBASE HPUX LAN"
*
* Login to the host system
*
LOGON wbtest
*
* Connect to the Database Environment on the remote system
*
SETDBE manufact/partsdbe
REMOTE
*
* Issue the SQL statement to get the part and lot information
*
SQL "SELECT PARTNUMBER, PARTNAME, LOTNUMBER, CONTRACTDUEDATE, &
            CONTRACTAMOUNT, CONTRACTNUMBER, JOBTYPE, RETENTION, &
            CONTINGENCYFEE, COSTMETHOD, JOBMANAGER, JOBSTATUS, &
            STATUSDATE, ESTSTARTDATE, ACTSTARTDATE, ESTCLOSEDATE, &
            CUSTOMERNUMBER, COMPANYNAME, INCOMEACCT, ACCRECACCT, &
            COMPLETED, COMPLETEDATE, ORIGBUDGET, REVBUDGET, &
            PROFITEST, PROFITBUDGET, PROFITJTD, CONTRACTAMTREV, &
            REVDUEDATE, LASTREVDATE, JTDACTCOST, JTDDIRECTCOST, &
            BILLABLEAMT, AMTBILLED, AMTRECEIVED, OVERUNDERBILL, &
            LASTBILLDATE &
     FROM   PARTSTABLE, LOTTABLE &
     WHERE  PARTSTABLE.PARTNUMBER = '!INPUTPARTNUM' AND &
            PARTSTABLE.PARTNUMBER = LOTTABLE.PARTNUMBER;"
OUTPUT_TAB
PRINT_REPORT partlot.rpt
MAIN
EXIT
```

How Did the Customer Use The Command File?

As soon as the service representative realized that the customer was inquiring about a part number, he or she would select the PAM menu choice that would run Information Access and automatically execute the command file. Depending on whether the user had been running another program or had just finished looking up a part number, between 4 and 7 seconds would pass before the command file would come up and ask the user to enter a part number.

The representative would enter the part number, press (Enter), and then continue asking the customer for additional information, such as their **customer number**, **company name**, **order number**, and order **date**, while Information Access was requesting the query from the ALLBASE/SQL program interface. As soon as the first lot number rows were read, they would be passed through a host file down to the PC, where they would be formatted and printed.

The last two pieces of information requested of the customer were superfluous - however, if the printout hadn't started by this point, asking for this data allowed the user to stall for a little more time while the query continued and the printout began.

Note

It should be noted here that, unlike the TurboIMAGE code, the SQL access code for Information Access will begin the output of data immediately upon finding it instead of waiting until all records have been found. This feature, combined with the "stalling" technique mentioned in the previous paragraph, provided a printout in as little **as 15 seconds**, and most always in a **20-40 second** timeframe starting with the point at which the user typed in the part number.

This almost invariably satisfied the needs of both the customer and the service representative.

As soon the output to the printer was finished, the user's PC would return to the PAM menu where he or she could immediately execute the part number lookup menu selection again.

What Weren't We Able to Accomplish?

There was one specific request from the users that we wanted to implement from the beginning, but were unable to do so because of a limitation within Information Access.

Most users envisioned the original solution as being very similar to our results above with one exception - they wanted the output to go to the screen rather than the printer. They had hoped to see all of the lots for the part number entered, one per screen, with the ability to (Page Up) and (Page Down) between each lot, finding the one(s) they were looking for very quickly.

Although this can be done interactively, when Information Access is run in "Batch Mode", it assumes that your formatted records will be placed either in a file for later printing, or output directly to the printer at run time. If you want to output to the screen in "Batch Mode", you have to give up the formatting.

Fortunately, as mentioned in the following section, we were able to accomplish this through other means.

How Do You Improve on a Good Thing?

Once we able to prove the validity of this approach and its ease of use, a separate development in the company coincided with ours to produce an even better solution.

The company had been in the process of transitioning some users into the HP NewWave environment and wondered if we could somehow combine the two and produce a more integrated solution that was even easier to use.

It turned out that with a little more effort, we were able to work out an integrated HP NewWave - Information Access setup that users found very easy to use.

How Do You Integrate HP NewWave and Information Access?

The same concept of using an Information Access command file to request a part number from the user, query the database for the proper information, and display it for the service representative was used in our HP NewWave solution.

We set up the command file as an icon on the HP NewWave desktop, so that as soon as the user "clicks" on the icon, the Information Access command file is executed.

We had to modify the command file slightly to output to a file instead of the printer. The new command file (without comments) looked like the following:

```
INPUT "Please enter part number:" inputpartnum
SET_CONNECTTYPE "ALLBASE HPUX LAN"
LOGON wbtest
SETDBE manufact/partsdbe
REMOTE
SQL "SELECT PARTNUMBER, ...
     FROM   PARTSTABLE, ...
     WHERE  PARTSTABLE.PARTNUMBER = '!INPUTPARTNUM' AND ...
OUTPUT_TAB

FILE_REPORT partlot.rpt TO \access\tmpfiles\partlot.pc8 OVERWRITE

MAIN
EXIT
```

There are two things to notice in this command file - both of them in the FILE_REPORT line towards the end.

- We are now FILING the report instead of printing it. The same report layout file (PARTLOT.RPT) is being used, so the output looks the same as the printed report. We are also able to use the same file each time (\ACCESS\TMPFILES\PARTLOT.PC8) by using the **OVERWRITE** option.

- We are writing to a file with the .PC8 extension.

Finally, we are using Information Access that has been installed into the HP NewWave environment. There is no difference in the software between the "standalone" version and the HP NewWave installed version of Information Access. However, when Information Access has been installed as part of HP NewWave Office or as part of the regular HP NewWave installation, it will be registered as a NewWave tool, or become "encapsulated". This means that objects created by the "encapsulated" Information Access will become NewWave objects themselves.

Note

The above prerequisites are important, because if they have all three been met, when the user "clicks" on the Information Access command file and enters the part number, when the command file finishes and the user is returned to the HP NewWave desktop, a new icon will appear that represents the object that was output from the FILE_REPORT command - in this case, the file called **partlot.pc8**, which contains the formatted output from Information Access.

How Did This New Solution Change Things?

The integration of the Information Access command file with the HP NewWave environment modified the steps taken by the service representative when a customer called inquiring about a part number.

Now he or she would take the following steps:

- "Click" on the Information Access Command File Icon from the HP NewWave desktop
- Enter the part number given by the customer
- Ask the customer for more information while the results are being output to a file and registered as a NewWave object
- "Click" on the new icon that represents the output from the command file
- Using NewWave Write, find the right lot number on the screen and give the requested information back to the customer
- Exit NewWave Write and "drag" the icon to the "trash bin" - no paper wasted

Was Everybody Happy?

Although the NewWave integration made a lot of users more friendly towards the system, you'll notice from the previous "steps list" that the user still needed to do a 2-step process - "click" on the Information Access Command File Icon, and then "click" on the new icon representing the object created by the command file. The programming support staff hoped that this sequence could be combined into a single command.

Even this solution was available through NewWave using the Agents capability. At the time the solution was initially presented, the

Agents capability was not yet implemented in the current release of NewWave. However, the customer was pursuing getting a Beta copy and whenever they manage to get the **Agents** functionality, they will have a one-step solution that provides them with everything they originally wanted - easy to use but powerful enough to satisfy the users and customers.

The customer service representatives, and especially the product-line managers were very pleased with the results. We were too.

TITLE: What's Behind Those Windows

AUTHOR: John Stachowisk
NW Tri County Inter. Unit
252 Waterford St.
Edinboro, PA 16412

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 4133

Writing Machine-Independent C Programs on the HP3000

Joseph Wm. Berry
Dataform Corporation
1498-M Reisterstown Rd.
Suite 272
Baltimore, MD 21208
301-764-5668

Introduction

I began writing C language programs a number of years ago when Hewlett-Packard introduced the HP-150 personal computer. My first project was a copy utility that copied complete diskettes. This program also displayed files and disk sectors in ASCII and hexadecimal. While I made many mistakes writing that program, I accomplished my main goal, learning C.

Since then, I have written many other programs and with each one my C skills have improved. I have found that only through practice will your C programming skills improve. You can read all the books about C that you want but you'll only begin to really know the language when you write code. It wasn't, however, until later in time that I learned about writing machine-independent C programs.

Two years ago, I designed a project with "porting" in mind. This program was going to be written on a PC and later transferred to the HP3000. The Oracle Corporation had been beta testing their new relational database management system for the HP Series 900 Spectrum computers. I had, for quite some time, been experimenting with HP's HPSQL and was eager to continue in the area of relational databases. Oracle claims to have the widest available number of platforms for their product including a full-blown system that ran on a PC. Their developers' package convinced me that I wanted to write an application using Oracle since I would be able to test out my Oracle interfacing on my PC at home. My project was to write a program, which I called IMAGICAL, that would convert Image databases to Oracle databases. I would develop it on a PC (which I had at home) and port it to the HP3000 (which was not as readily available, especially late at night).

Since this program was supposed to read Image databases, I needed this facility running on a PC. I tried to contact a company that claimed to have Image running on PCs. I wanted something that was compatible to MicroSoft C for two reasons: (a) it was the closest thing to a standard C compiler that I was aware of; (b) Oracle required either MicroSoft's C or Lattice's C in order to run. Unfortunately, I never got to first base with this company. They didn't bother answering my mail. When I ran into them at an Interex conference, I raised the question. They promised to give me an answer about their compatibility but never did. Obviously, I sought an alternative solution.

A second major concern about my new product was terminal display. A PC monitor has its own set of escape codes (from ANSI.SYS) that is different from HP terminals. This program was to run on an HP terminal, yet I was developing the product on my PC. This too required resolution.

Image on a PC

I was not about to develop a real Image database system on my PC in order to develop and test my database conversion program. All I really needed was a procedure or set of procedures that would act as an interface to Image. This turns out to be one of the most important rules for developing computer-independent C code: encapsulate any procedure or technique that requires direct access to the host's operating system. Since Image is a DBMS that is platform dependent (that is, it only runs on HP3000s), encapsulate it. And this, I did.

I treated the Image datasets as individual tables. I needed a procedure that would return to me the names of the tables defined in the database. I also needed a procedure that would return to me the characteristics of any given table (that is, the data item names and their sizes). I decided to create a flat file "pretend" Image database.

This database would really consist of a number of flat files that contained not only the descriptions of the datasets and data items but also actual data which I would create with a utility program. Since I didn't know how many datasets there would be in a particular database, I chose to use C's dynamic storage allocation routines to define a chunk of storage for 199 datasets. Then, after ascertaining how many datasets actually existed in the database, I "down sized" the data buffer. The following structure was used to define each dataset.

```
struct dbset_info
   {
   char                name[16];
   char                type;
   int                 num_items;
   long                capacity;
   long                entries;
   struct dbitem_info  *items;
   };
```

A separate data buffer was defined that contained descriptions of the data items associated with a particular dataset. The structure for the data item is shown here:

```
struct dbitem_info
  {
  char                  name[16];
  char                  type;
  int                   size;
  int                   subcount;
  int                   start_pos;
  };
```

Note in the appendix a "generic" database open procedure. When I eventually ported this program to the HP3000 (and afterwards to other platforms) the calls to this procedure (as will as the code in this procedure itself) did not have to change. The real database open was performed in the procedure _open_db. On the HP3000, this procedure used calls to DBINFO to fill the data structures with information about the database. On the PC, I opened a normal file that contained the list of datasets as well as their types (automatic master, etc.) and number of entries. Note the differences between the two versions of _open_db as they appear.

Terminal Handling

As I mentioned earlier, terminal handling is an important factor in writing machine independent code. You want your program to run on a variety of terminals and monitors. Taking advantage of HP's escape strings to enhance terminal output makes sense when you know you will only be running your program on an HP terminal. However, more and more users are now connecting PCs and other kinds of terminals to their HP computer system.

While developing this utility, I knew nothing about UNIX nor UNIX's solution to terminal handling (I'll discuss this shortly.) I set out looking for a generic solution to the problem of writing procedures that would allow access to different types of terminals and monitors. I found my solution in the book, C Programmer's Library by Purdum, Leslie, Stegemoller, Que Corp., 1984. This book contains a complete solution to the problem of terminal handling. The authors wrote a number of user callable procedures that applications utilize when manipulating the terminal.

The procedures include CLRSCR, for clearing the screen; CURSOR, for moving the cursor to a specified row and column; and ERAEOL and ERAEOP, for clearing the line and screen from the current cursor location.

Some of the functionality that I wanted was missing, however, and so I added a number of procedures, such as GETCURSOR, for inquiring as to the current row and column position of the cursor.

Porting this subsystem to the HP3000 was the hardest part of converting IMAGICAL. The most difficult part of the problem was getting the GETCURSOR procedure to work. The main symptom was that the HP terminal locked up and had to be reset. I was unable to isolate the problem until I put a data scope on the line between the HP3000 and the terminal. That is when I learned about the FSETMODE intrinsic that can turn off the carriage return / linefeed that MPE sends to the terminal after a read has been completed. This is settable by entering a "4" in the mode parameter to the FSETMODE call. See the Intrinsics manual for a better understanding of the problem.

The second problem was the read trigger character that didn't exist when I tried to read the cursor position from the terminal. The data was sent too quickly for the program to turn the write around and issue a read. Intrinsics FCONTROL and FDEVICECONTROL were used to solve that problem (see the following code and refer to the Intrinsics Manual).

The C language does not contain standard procedures for easily dealing with unbuffered I/O. This problem becomes readily apparent with the following example:

```
puts("Enter your name: ");
gets(name);
```

You want the cursor to remain on the same line as the prompt. This code works fine on a PC but is totally inadequate on the HP3000. Since the HP3000 buffers all of its I/O, executing the above two statements produces a cursor waiting for input without any sign of a prompt string. After the user enters the name and hits the carriage return, the "Enter your name: " prompt will appear.

To develop a generic solution, I took advantage of MicroSoft's C compiler, which contains a set of console functions similar to the standard I/O routines. For example, there is a procedure called cputs which immediately sends the string to the monitor. None of the C compilers on the HP3000 have a similar routine. I therefore used that procedure for all of my output and wrote my own version of cputs for use on the HP3000. I used the HP intrinsic FWRITE with a carriage control value of %320 to force writing without issuing a carriage return and line feed.

```
int cputs(string)
char *string;
{
  FWRITE(mpe_stdout, string, -strlen(string), 0320);
}
```

A similar problem existed with the single character reads that I needed to perform. MicroSoft has a pair of functions, getch and getche, that return single characters. Getche echoes the character after you type it in. Getch does not echo the character (useful for requesting passwords, for example). None of the

C compilers on the HP3000 has such a function and so I wrote one that would be compatible to MicroSoft's.

```
int getche()
{
/*
   This routine retrieves a single character typed on the
   keyboard. The character is echoed on the screen.
*/
   int  len;
   char data[4];

   len = FREAD(mpe_stdin, (unsigned short *) data, -1);
   return(len = data[0]);
}
```

```
int getch()
{
/*
   This routine retrieves a single character typed on the
   keyboard. The character is not echoed on the screen.
*/
   int  len;
   char data[4];
   int  dummy;

   FCONTROL(mpe_stdin, 13, &dummy);    /* turn echo off */
   len = FREAD(mpe_stdin, (unsigned short *) data, -1);
   FCONTROL(mpe_stdin, 12, &dummy);    /* turn echo     */
   return(len = data[0]);
}
```

Terminal Handling with UNIX

Once everything was working on the PC and the HP3000, I decided to make the big move to UNIX. UNIX is not just another operating system. It is another world. I have learned that in order to survive the world of UNIX, a fair amount of time must be invested. Alternatively, it requires help from a UNIX guru. UNIX already has terminal handling capability - it is called curses. Curses consists of a number of procedures for moving the cursor and generating windows. This package should not be confused with the popular X-windows. X-windows requires a graphics terminal while curses works with virtually any kind of terminal.

My initial thought had been to port my own procedures to UNIX. The idea was that since I succeeded in porting my procedures to the HP3000, I should be able to do the same port to UNIX. This, unfortunately, turned out to be more difficult than I had anticipated. To modify the input and output file types so that I could perform single character reads and writes did not turn out to be so easy. I experimented with procedure IOCTL (UNIX's version of FCONTROL) but could not get the I/O to work the way I wanted it to.

I then replaced my routines with the curses routines by using a number of #define statements. For example,

```
#define getche          getch
#define gets            getstr
#define cursor(r, c)    move(r, c)
#define cputs(x)        addstr(x); refresh()
```

A couple of procedures actually required modification, but that work was accomplished in under 30 minutes.

In retrospect, knowing that there is a package available like curses, I would have taken advantage of it and minimally used its syntax. Using a standard, any standard, is usually better than forging ahead alone.

Messages

The HP3000 has a mechanism for creating and displaying text messages that are stored independently of the program. These messages are stored in a special file by set number and message number within the set. In addition, up to five numeric or alphanumeric parameters may be defined in the message. These parameters are automatically imbedded into the text, if present. To programmatically retrieve a message, a call to procedure GENMESSAGE is invoked. This procedure returns the message after substituting any parameters that were passed to it. The message catalog file is created using an HP utility called MAKECAT.

I have always liked this technique of handling messages and wanted to use it in IMAGICAL on the PC. The problem, of course, is that there is no GENMESSAGE procedure on the PC. I solved the problem the "easy" way. I wrote my own genmessage procedure and MAKECAT program. This software is almost identical in functionality to HP's. When I ported my software to the HP3000, I was about to scrap my routines and use HP's. However, since I knew they worked, I simply ported them to the HP3000 and continued to use them instead of HP's routines. Porting them to UNIX also proved to be a trivial task requiring only changes in the #include statements.

Standard I/O

Besides terminal handling, the most difficult part of the conversion process was standard I/O handling, that is, creating and accessing files. Each C compiler and each hardware platform had its own idiosyncrasies.

I decided to use what is called in the UNIX world as "low level" I/O. Essentially, this means that I use procedures that indicate the lengths of the records that I want to read and write. These procedures have names such as OPEN, CREAT, READ, WRITE. Such procedures are very reminiscent of HP's FREAD and FWRITE intrinsics. The procedures exist on the PC in MicroSoft's C compiler. They also exist in HP's C compiler for the XL systems. However, neither Tymlabs nor CCS define those procedures in their respective compilers. What makes things a bit worse is that some of the parameters on the OPEN statement were different between HP's C compiler and MicroSoft's C. After some hassle, I chose to ignore HP's implementation of these low level routines and wrote my own. The appendix presents an example of my implementation of the READ procedure.

A certain oddity needs to be clarified about this READ procedure. The HP I/O intrinsics require that the buffer be of type integer while the standard low level routines use byte pointers for their buffers. A problem will occur if the character buffer is not word-aligned. I learned this the hard way when I kept losing a character during my reads. That is why the actual buffer used in the FREAD intrinsic is of type "int" and the data is then transferred via the MEMCPY procedure to the user-requested buffer.

Comments about the HP3000

Using the HP3000 as a development platform can be a very frustrating experience. Most of my HP3000 work has been on a small series 925 Spectrum system. When I used HP's HP-C compiler, I marvelled at the fast compile times. That wasn't due to the C compiler, however, but rather due to the fast processor on the series 925 running in native mode. I performed a similar test using compatibility mode versus native mode Fortran and came up with similar results.

I have recently been using Tymlabs' C compiler on a series 925 computer and have managed to read many magazines waiting for my programs to compile. This compiler runs in compatibility mode and it is this "feature" that makes my compiles crawl like a snail. I have not run any tests of CCS's compatibility mode C compiler but I suspect that it will also be significantly slower than HP's native mode compiler.

If you are writing a program in C that will need to run on both the classic and Spectrum series of HP computers, then you will obviously need a compatibility-mode C compiler, that is, a compiler that runs on the classic systems. Your choices are C compilers by Tymlabs or CCS. Nevertheless, if the development platform will be a Spectrum system, having HP's native mode C compiler available to speed up the development cycle may be very prudent.

On the other side, however, is the unfortunate fact that HP's compiler does not yet support the new ANSI standard. It has no prototyping available, a feature I have come to depend upon for writing clean code. Both MicroSoft's, Tymlabs', and CCS's C compilers offer prototyping. When I ported my program to UNIX, I learned that its compiler, too, does not offer prototyping. An available C compiler on UNIX called the Gnu C compiler does have this feature; I have not yet acquired it.

General Comments and Conclusions

For some strange reason, the routine most frequently needed and usually missing on the various hardware/software platforms was ITOA, the function that converts an integer number to its ASCII equivalent. It exists on MicroSoft C but on NO other compiler that I have used. A very usable source code example can be found in Kernighan and Richie's book, The C Programming Language. By extension, the function LTOA can be derived from it.

The C programming language allows for the creation and use of global variables. These variables can be defined in one source file and referenced in another source file. As long as all the source files are PREPed together to form the executable program file, all three C compilers on the HP3000 are able to deal with the global variables. In IMAGICAL, I had built up my own terminal handling package that required some global storage for identifying the terminal type and its characteristics. These routines are typically maintained in a library module (using classic HP3000 terminology, an SL file). MicroSoft's C had no problem dealing with library files containing globals. HP C also had no problem using the new LINKEDIT commands. Tymlabs' C, however, could not store globals into its library. This is due to the fact that it relies on HP's SEGMENTER for library maintenance. And HP's SEGMENTER does not permit any global storage. To get around the problem, I had to completely discard the use of libraries for maintaining frequently used code. I was unable to test out CCS's C compiler but since they do not use the SEGMENTER, but rather their own link editor, I assume that their operating environment would permit the use of globals in library files.

There are general comments that can be made about writing programs that are portable. One of the most important rules is to never hard code any constants into the program. Wherever constants are required use #define statements, preferably in a user-defined header file. In the case of my terminal handling code that was ported to UNIX with curses, I placed my new #defines (see above) in my terminal header file, TERMINAL.H. Since all the procedures that performed any terminal I/O already included this header file (#define "terminal.h"), my changes only required a recompile to take effect.

Make sure that your compiler includes an appropriate #define variable so you can localize your code based upon the compiler. HP's C compiler defines a variable called "mpexl". MicroSoft's C compiler has "MSDOS". I couldn't locate one for Tymlabs' C, so I created my own. I simply added the statement "#define HP3000" to my main header file.

Never code MPE intrinsics into your main C application if you want to have machine independent code. Isolate such requirements into specific stand-alone procedures. For example, in one application, I needed a mechanism for creating son processes. This code was to be executed in two different places. Instead of calling the HP intrinsic, CREATEPROCESS, I wrote a MicroSoft C compatible procedure called SPAWNL.

The important rule is to always be thinking about the software port that needs to be made. If you code your program with this port in mind, performing the actual conversion will be much easier.

open_db

```
#define MAX_DSNUMS 200

int open_db(dbname, password, mode, dbhandle, dataset_ptr)
char *dbname;
char password[];
int  mode;
int  *dbhandle;
struct dbset_info **dataset_ptr;

{
char    error_buf[80];
int     i;

/* Allocate maximum amount of memory for all possible datasets
   -- afterwards, get rid of any excess memory.
*/

num_datasets = 0;
*dataset_ptr = (struct dbset_info *)
                malloc(MAX_DSNUMS*sizeof(struct dbset_info));
if (*dataset_ptr == NULL)
   {
   strcpy(error_buf, "Not enough memory available in OPEN_DB malloc.");
   display_error(error_buf);
   *dbhandle = 0;
   exit(1);
   }


/* Perform the real dbopen */

_open_db(dbname, password, mode, dbhandle, dataset_ptr, &num_datasets);
if (*dbhandle == 0)
   {
   free(*dataset_ptr);
   *dataset_ptr = NULL;
   return(0);
   }


/* We now get rid of the excess storage that we allocated since we now
   know how many datasets there were in that database.
*/

if ((*dataset_ptr = (struct dbset_info *) realloc(*dataset_ptr,
          (num_datasets+1)*sizeof(struct dbset_info))) == NULL)
   {
```

```
   *dbhandle = 0;
   strcpy(error_buf, "Error #1 in REALLOC function in ..");
   display_error(error_buf);
   exit(1);
   }

}
```

_open_db

```
_open_db(dbname, password, mode, dbhandle, ds, num_ds_sets)
/*
This is the real database open routine that is executed on the HP3000.
*/
char *dbname;
char *password;
int  mode;
int  *dbhandle;
struct dbset_info **ds;
int *num_ds_sets;
{
   short  int base_name[30];
   short  int my_password[5];
   short  int status[10];
   short  int qualifier;
   short  int dsnums[MAX_DSNUMS];
   short  int dsbuf[18];
   struct dbset_info *ds_ptr;
   int    i, j;
   int    r, c;
   short  int my_mode;


   my_mode = mode;
   strcpy((char *) &my_password[0], password);
   strcat((char *) &my_password[0], ";");     /* add deliminator */
   strcpy((char *) &base_name[0], "  ");
   strcpy((char *)&base_name[1], dbname);     /* save place for return code */
   strcat((char *)&base_name[1], ";");        /* add deliminator */

   DBOPEN (base_name, my_password, &my_mode, status);
   if (status[0])
      {
      image_error(status);
      return;
      }

                           /* identify all data sets available */
   my_mode = 203;
```

```
    DBINFO (base_name, &qualifier, &my_mode, status, dsnums);
    if (*status)
       {
       image_error(status);
       *dbhandle = 0;
       return;
       }

    *num_ds_sets = dsnums[0];
    if (*num_ds_sets == 0)
       {
       help_info(509, &r, &c, TRUE);
       *dbhandle = 0;
       return;
       }

    ds_ptr = *ds;
    my_mode   = 202;
    while (dsnums[0]--)          /* get info on each data set */
       {
       qualifier = abs(dsnums[*num_ds_sets - dsnums[0]]);
       DBINFO (base_name, &qualifier, &my_mode, status, dsbuf);
       if (*status)
          {
          image_error(status);
          *dbhandle = 0;
          return;
          }

       /* Save the various statistics about each dataset */

       memcpy(ds_ptr->name, (char *) &dsbuf[0], 16);
       memcpy((char *) &ds_ptr->type, (char *) &dsbuf[8], 1);
       ds_ptr->num_items = 0;  /* won't know this till next routine */
       memcpy((char *) &ds_ptr->capacity, (char *) &dsbuf[15], sizeof(long));
       memcpy((char *) &ds_ptr->entries, (char *) &dsbuf[13], sizeof(long));
       ds_ptr->items = NULL;
       ds_ptr++;
       }

    strncpy(ds_ptr->name, blanks, 16);

    *dbhandle = base_name[0];
}
```

_open_db

```
int _open_db(dbname, password, mode, dbhandle, ds, num_ds_sets)
char *dbname;
char password[];
int  mode;
int  *dbhandle;
struct dbset_info **ds;
int *num_ds_sets;
/*
This is the "fake" database open routine that I used on my PC. The
database name was actually the name of a normal file on the PC. Each
record contained the name of a dataset, its type, number of entries,
etc.
*/
{
   FILE *db_file;
   char *data;
   char data_line[80];
   char error_buf[80];
   struct dbset_info *ds_ptr;
   char *ptr;
   int  i;

ds_ptr = *ds;

if ((db_file = fopen(dbname, "r")) == NULL)
   {
   *dbhandle = 0;
   return;
   }

while ((data = fgets(data_line, sizeof(data_line), db_file)) != NULL)
   {
   (*num_ds_sets)++;
   strncpy(ds_ptr->name, "                ", 16);
   ptr = strtok(data, " ,");
   data = NULL;
   i = 0;
   while (isalnum(*ptr))
      ds_ptr->name[i++] = *ptr++;

   ptr = strtok(data, " ,");
   ds_ptr->type = *ptr;

   ptr = strtok(data, " ,");
   ds_ptr->capacity = atol(ptr);

   ptr = strtok(data, " ,");
```

Computer Museum

```
    ds_ptr->entries = atol(ptr);

    ds_ptr->items          = 0;                    /* defaults */
    ds_ptr->num_items      = 0;

    ds_ptr++;
    }

strncpy(ds_ptr->name, "                ", 16);
fclose(db_file);
*dbhandle = 1;
```

getcursor

```
int getcursor(row, col)
int *row, *col;
{
/*
  This function returns the current row and column location of the
  cursor on the screen.
*/
int i;
short int dummy, error_ret;
char return_str[16];
char *token;
char *ptr;

#ifdef MSDOS union REGS regs; #endif

   if (__termv.disable_use)  /* We may be running in batch mode.*/
       {                     /* In that case, there is no point */
       *col = 1;             /* in returning a row, column.     */
       *row = 1;
       return(-1);
       }

#ifdef MSDOS
                             /* cheat -- use DOS's BIOS calls   */
   regs.h.bh  =  0;          /* page number                     */
   regs.h.ah  =  3;          /* cursor position function number */

   int86(0x10, &regs, &regs);
   *row = regs.h.dh;
   *col = regs.h.dl;
   return(-1); #endif


   if (__termv.__curg[0])
       {

       if (__termv.__termdefn == 2 ||  __termv.__termdefn == 3)
          {                        /* EM 100 or VT 100 mode */

          dummy = (short int) 'R';

#ifdef mpexl
          FCONTROL (mpe_stdin, (short int) 25, &dummy);
          if (ccode() != CCE)
             {
             cputs("\n\rUnable to set terminal terminator for
stdin.");
             exit(1);
```

```
      }

   dummy = (short int) 'n';       /* set up for a new terminator */
   FDEVICECONTROL(mpe_stdin, &dummy, (short int) 1,
         (short int) 192, (short int ) 32,
         (short int) 3, &error_ret);
   if (ccode() != CCE)
      {
      cputs("\n\rUnable to execute FDEVICECONTROL.");
      exit(1);
      } #endif

   __send(__termv.__curg);        /* This procedure writes the  */
                                  /* contents of __termv.__curg */
                                  /* to the monitor.            */
   i = read(0, return_str, 8);    /* Read the results from the  */
                                  /* screen, including row, col.*/

   dummy = 0;    /* reset the end of line */ #ifdef mpexl
   FCONTROL(mpe_stdin, (short int) 25, &dummy);

   dummy = (short int) 17;        /* reset to DC1 */
   FDEVICECONTROL(mpe_stdin, &dummy, (short int) 1,
                 (short int) 192, (short int) 32,
                 (short int) 3, &error_ret); #endif


   *row = atoi(&return_str[2]) - 1;    /* Break out the row, col */
   ptr = &return_str[2];
   while (isdigit(*ptr)) ptr++;
   ptr++;                       /* skip over ; */
   *col = atoi(ptr) - 1;
   return(-1);
   }         /* end of EM 1000 & VT 100 */

if (__termv.__termdefn == 4)  /* IBM ANSI */
   {
   __send(__termv.__curg);
   for (i=0; i<9; i++)
      return_str[i] = getch();
   *row = atoi(&return_str[2]) - 1;
   *col = atoi(&return_str[5]) - 1;
   return(-1);
   }

if (__termv.__termdefn == 1)  /* HP generic */
   {
   __send(__termv.__curg);
   read(0, return_str, 11);
   *col = atoi(&return_str[3]);
   *row = atoi(&return_str[7]);
   return(-1);
```

```
            }

        return(0);
        }

    return(0);
}
```

read

```
#define ERROR_INDIC       -1
#define MAX_ALLOWED_BUF  1024

int read(fnum, realbuf, bufsize)
int  fnum;
char *realbuf;
int  bufsize;

{
   int buf[MAX_ALLOWED_BUF];
   int bufsizegot;
   short int errorcode;

   if (abs(bufsize) > sizeof(buf)) return(ERROR_INDIC);
   if (bufsize > 0) bufsize = -bufsize;

   bufsizegot = FREAD((fnum != 0 ? fnum : mpe_stdin), buf, bufsize);
   switch (ccode())
      {
      case CCE:
               memcpy(realbuf, (char *) buf, abs(bufsizegot));
               return(abs(bufsizegot));
               break;

      case CCG:
               return(0);
               break;

      case CCL:
               FCHECK(fnum, &errorcode);
               errno = errorcode;
               return(ERROR_INDIC);
               break;
      }
}
```

DESKTOP PUBLISHING FOR THE 1990's
WHAT'S NEW AND WHAT'S NEXT

DAVID SWINKIN
HEWLETT-PACKARD COMPANY
3 PARKLAND DRIVE
DARIEN, CONNECTICUT
06820

BOSTON - 1990
INTEREX HP USERS CONFERENCE
AUGUST 20-23

PAPER 4187

TABLE OF CONTENTS

I. INTRODUCTION

The 1990s is an exciting time for the "communications revolution". In less than 10 years, the personal computer industry has matured from humble beginnings with a handful of product offerings to a state where users expect uncompromised performance with 80386 and 80486 processors, unparalleled software power and flexibility and the highest quality output with laser printers. Many standards for software and hardware have emerged from large corporations such as IBM and Hewlett-Packard as well as smaller companies such as Aldus and Adobe Systems. Desktop publishing (DTP) has matured also. New and improved software and hardware is providing users with capabilities that were not possible a few years ago. DTP began from the need to create dynamic memos and documents and has become the standard methodology for business communications. DTP is as common today in the workplace as word processors, graphics, database and spreadsheet software. DTP software is exciting because it has the ability to easily integrate data files from virtually all personal computer software packages and create a visually exciting document.

DTP allows individuals and companies to design and produce their own newsletters, business reports, catalogs, manuals, trade journals, catalogs and price sheets at a fraction of the cost of conventional printing methods involving professional printing operations. DTP allows the user to maintain complete control of the publishing cycle including design, schedules and last minute changes. By integrating many software and hardware components, a user can produce dynamic and professional documents very quickly. Anyone with a personal computer and laser printer can participate in the DTP process as long as they are willing to learn design techniques and are not afraid to experiment with the powerful hardware and software products currently available in 1990.

In the pages that follow, the methodology for integrating the exciting hardware and software components of 1990 into a successful MS-DOS(R) based DTP workstation will be presented. In addition, products that enhance the DTP environment will be introduced. These products provide enhancements for the casual as well as advanced DTP user. The final objective of this paper will assist the DTP user in understanding where the technology is going in the 1990's in order for the DTP user to understand what to expect in the near future.

II. DESKTOP PUBLISHING IN 1990

Dynamic looking desktop publishing documents produced by in-house personnel has changed the workplace for everyone. The technology that powers DTP software and hardware has matured rapidly, making 1990 an exciting time to work in the DTP arena. The technology available today allows us to implement DTP systems that produce exciting documents, increase productivity in the workplace and control the entire printing process. A beginner DTP solution can be implemented with a low cost 80286 based personal computer and a laser printer and simple, low cost DTP software that total $5,000 or less. As hardware performance, software performance and user expectations increase, the price of the DTP installation can go up dramatically.

By implementing an in-house DTP solution, a company can expect to save $113.00 (or more) per page over professional typesetting activities including layout, design, typesetting, proofreading, paste-up and camera ready artwork. Aldus Corporation (author of a top selling DTP software package) estimates that a 16 page newsletter produced monthly can cost a company over $23,500.00 per year using professional outside printing services. That same newsletter produced by in-house DTP personnel will cost less than $2000.00 per year. Besides the cost, a 16 page camera-ready newsletter can be produced in under four days versus nine days using outside services and allow complete control of the publishing cycle. Another important benefit of DTP allows for a company to maintain complete control over confidential material.

The most important factor for a successful DTP installation is setting the users expectations regarding system performance and software capabilities. An 8 megahertz (MHz) personal computer cannot compare to the performance of a 20 MHz 80386 personal computer. Although this is perhaps an obvious statement, the user must know what their budget is and put more money in the one or more DTP components that will specifically satisfy their needs and expectations. A faster printer with additional font capabilities would satisfy a user looking for sophisticated output more so than investing in a faster, more expensive personal computer.

There are many components that make up the DTP systems of 1990. Besides a personal computer and laser printer, a user has many options to choose from including:

o...Scanners and image editing software
o...Cartridge fonts and soft fonts
o...Page makeup and word processing software
o...Presentation graphics, paint and draw software
o...Monitors including high resolution, full-page and two-page
o...Operating environments
o...Black and white or color output
o...OCR software

After the introduction of an entry-level, mid-range and high-end performing DTP workstation, these components and options will be illustrated in the next chapter. The price point for an outstanding entry-level system will be $12,000 and will increase to $20,000 for the mid-range and $25,000 for the high-end solution.

A user can at any time upgrade individual components to reach the next performance level. As the user becomes more aware of what the technology has to offer, the potential of "desktop" publishing can be realized. Specifically, their output can easily be measured against the output from professional printing operations.

THE ENTRY-LEVEL SOLUTION

A desktop publishing system can be successfully implemented at many price points. By simply adding desktop publishing software to an existing personal computer that has access to a laser printer, a user can begin to "publish" in-house documents. These documents would be limited because many other components are needed to create visually exciting documents. An entry-level DTP user with hardware and software in this category is capable of producing simple newsletters and multicolumn reports.

Lets begin with the personal computer. An entry-level solution in 1990 begins with a 12 MHz, 80286 personal computer. The definition of entry-level implies the machine has acceptable performance for demanding software applications. Although most top-rated desktop publishing software will operate in a standard 640K MS-DOS environment, adding an additional megabyte of memory is the best and lowest priced option for improving performance. The lowest acceptable video solution is a VGA adapter card with a monochrome VGA monitor. Desktop publishing on a personal computer requires a parallel port for faster data communications to the printer, an industry standard mouse, and at least one floppy disk drive for document archiving and hard disk backup. Finally, the minimum recommended hard disk capacity is 40 megabytes since a desktop publishing document and related data files can easily exceed one megabyte or more.

The laser printer is a standard tool for the in-house desktop publishing industry and will produce text output at 300x300 dots per inch. In fact, it was the introduction of the low cost, desktop laser in 1983 that allowed DTP to mature in the workplace. There are low cost desktop laser printers available today that will do a good job for low volume printing requirements (i.e. 80 DTP pages per month). It is interesting to note that the technology available in 1990 offers laser printers that do not go up <u>dramatically</u> in price as performance increases. The price difference between top rated laser printers rated at four pages per minute versus eight pages per minute is about $1000.00. A four page per minute, 300x300 DPI laser printer will yield the same <u>quality output</u> as an eight page per minute, 300x300 DPI laser printer but at a slower speed. To produce visually exciting documents on a limited budget, it would be better to go with a slower laser printer and purchase accessories such as fonts or additional memory. Most laser printers come with 512K of memory. To produce documentation with integrated text, graphics and perhaps scanned images, it will be necessary to include an additional one or two megabytes of printer memory. This decision depends on whether you will be choosing fonts that are cartridge based or disk based. (Cartridge fonts are commonly referred to as "hard fonts" and disk based fonts are commonly referred to as "soft fonts".) Cartridge fonts require no additional printer memory and are easier to use but are typically more expensive. Disk based fonts provide more flexibility (i.e. more font styles) at lower cost but require more printer memory and computer disk space to store the fonts. Fonts are chosen based on your need and desire to print DTP documents with different styles and heights of characters.

There are many other accessories available for the desktop laser printer. As a user becomes more sophisticated and knowledgeable about the DTP market, options such as page description languages (i.e. Postscript from Adobe Systems or PCL from Hewlett-Packard), printer sharing devices, 1000 DPI hardware upgrades and accessory boards, similar to expansion cards that fit into a personal computer should be investigated to improve output capabilities and enhance personal productivity.

Scanners provide the DTP user with an extra tool to create dynamic, visually exciting documents. Scanners can be used as OCR devices for text integration or as a method to integrate line art or graphics into your final document. Many software utilities allow desktop scanner users to "touch-up" or even alter an original graphic object. Scanners require the use of an expansion slot in your personal computer and can be added at any time to your DTP workstation when your budget allows. The OCR ability of a scanner would provide the user with the opportunity to save time when integrating documents not produced on the DTP users' workstation. There are many OCR software packages available today at different price levels, beginning at $300.00.

Software products for desktop publishing are numerous and address many levels and needs. Most DTP software are sold as tools to integrate text and graphics from other software vendors. There are currently two software packages that dominate the MS-DOS based DTP environment. These packages can be used by casual as well as advanced users very successfully. Aldus Pagemaker and Xerox Ventura Publisher are full featured, DTP packages that integrate virtually all industry standard file formats. These software packages are not intended to be used as text processors or graphics programs but rather as a tool to integrate files from word processing packages such as Wordperfect or Microsoft(R) Word, and graphics packages such as Lotus Freelance, Harvard Presentation Graphics or Hewlett-Packard Graphics Gallery. The choice between Pagemaker and Ventura is difficult because each is outstanding and offer similar capabilities. Pagemaker is considered a better choice for documents of less than 100 pages and is easier to learn making Pagemaker a better choice for the casual DTP user. Ventura is better at longer documents and has more advanced features. These software products provide an excellent solution for desktop publishing in 1990. For a casual user, there are many alternate DTP software products that do not offer the full array of features that Pagemaker and Ventura offer and are priced at typically less than $300.00. (These products are discussed in Appendix 1)

The choice of word processors and graphics products are an integral part of the DTP solution. The choice of which product to select is not in the scope of this paper and is best left to the individual. Software decisions are based on specific requirements by the individual and are sometimes dictated by the company the individual works for. Before purchasing additional software, it is important to determine that the new software is compatible with your DTP environment.

For illustration purposes, software that is considered to be an industry standard or "best in class" by various industry sources has been selected. The following illustrates the components of an entry-level DTP workstation in the MS-DOS environment:

HARDWARE:	**US LIST PRICE AS OF 3/90**
HEWLETT-PACKARD ES/12 80286, 12 megahertz clock speed 640K main memory 5.25 inch floppy disk 42 megabyte (MB) hard disk VGA adapter One serial, one parallel port 7 I/O slots Desktop design	$3,299.00
MS-DOS 3.3	135.00
VGA monochrome display	349.00
Mouse	155.00
Parallel cable	55.00
Hewlett-Packard Laserjet IIP	1,495.00
Laserjet IIP memory 1 megabyte memory upgrade	495.00
Hewlett-Packard Microsoft "Z" font cartridge	250.00
Hewlett-Packard Scanjet Plus scanner with interface kit	2,190.00

SOFTWARE:	**US LIST PRICE AS OF 3/90**
PC Environment: Microsoft Windows 2.11	149.00
Word Processing Software: Wordperfect Version 5.0	495.00

ENTRY-LEVEL CONFIGURATION CONTINUED...

Graphics Software:	
Harvard Presentation Graphics	495.00
Desktop Publishing Software:	
Aldus Pagemaker Version 3.0	795.00
-or-	
Xerox Ventura Publisher, 2.0	895.00
OCR Text Scanning Software:	
ReadRight	495.00
Pagemaker solution:	**$10,852.00**
Ventura Solution:	**$10,952.00**

CONFIGURATION SUMMARY:

This configuration features a 12 megahertz, 80286 personal computer and will allow a user to create text and graphics files with industry standard software and integrate these files into a full featured DTP software package. The output will be produced on a laser printer rated at four pages per minute. With the additional printer memory, the user will be able to print a full page of text or graphics at 300X300 dots per inch. The additional font cartridge will produce helvetical and times roman typefaces including bold and italic up to 14 points (72 point equals one inch). Finally, the scanner will allow the user to integrate graphic images or line art into the DTP software and read good quality printed material into a wordprocessor with the OCR software for integration into a DTP document.

THE MID-RANGE SOLUTION

As your needs for more sophisticated output grow it is important to become familiar with the advanced options and accessories that the technology of 1990 provides. A user that will be producing extensive, multi-column business reports or proposals, newsletters with extensive graphic needs or long DTP documents will need additional computing power. This power will be used to process your DTP documents faster. In addition, a faster laser printer with extensive font capabilities will offer the DTP user the flexibility required to produce sophisticated output.

Many PC platforms address the need for the additional computing power required at this level. Prices for a system that uses the popular 80386 processor at 16, 20, 25 or 33 megahertz (MHz) do not differ substantially. Generally there is a $2,000.00 price difference between a 20MHz and 25MHz machine. A user will enjoy a substantial gain in productivity by using a 25MHz PC rather than a 20MHz PC. Perhaps more important is the fact that the additional cost of the higher priced 25MHz systems usually includes larger disk capacities. Vendors typically configure a 16MHz 80386 machine with a 40 megabyte hard disk versus a 100 megabyte (or more) hard disk on the faster configurations. The larger disk capacity will allow the DTP user to have access to more on-line data and provide faster disk access time allowing the software to perform at its best.

There is another alternative for a PC platform using an 80386 processor. The 80386SX processor offers complete compatibility with the 80386 processor and typically provides a $2000.00 cost savings. The 80386SX does not offer the same performance of an 80386 but does offer a substantial performance increase over a 12MHz 80286 processor. Remember, the speed and computing power of a users PC will not affect the appearance of the users laser printer output. The DTP user must balance their need for computing power with the more important need to purchase the accessories that will produce the desired output.

There are two industry standards regarding the method that DTP software communicates with a printer. Printer Command Language or PCL is a standard developed by Hewlett-Packard Company and adapted as a defacto "PC environment" standard in 1983 with the introduction of the original Laserjet printer. Another standard is Postscript from Adobe Systems. Postscript is the defacto standard on non-MS-DOS platforms but shares the MS-DOS platform for DTP output with the Laserjet family and PCL. Postscript offered more defined functionality than the 1989 version of PCL. However, Postscript functionality added $2000.00 or more to the price of the printing solution. In 1990, Hewlett-Packard announced the Laserjet III and released a new version of PCL (PCL Level 5) that included many options that were previously available only with Postscript. Most notably, PCL Level 5 now allows the user to scale fonts that are built into the printer or downloaded from disk based fonts. PCL Level 5 allows up to 999.75 point or approximately 14 inch characters. In addition, capabilities such as reverse print, text shadowing, rotating and angling text are now standard with the new HP Laserjet III with PCL Level 5. The importance of this product to the DTP environment cannot be understated. Capabilities that existed only on Postscript printers that cost upwards of $4500.00 are now available for $2395.00. The other important contribution of the new Laserjet III is a patented technology called "resolution enhancement" that allows the Laserjet III to compare to a 600 DPI printer. This technology alters the size and location of the dots placed on the printed page. By allowing variable dot sizes, the edges of letters and graphics print with no jagged edges. The Laserjet III is perhaps the most important new product announcement in 1990 in the DTP marketplace!

In 1990, Postscript still has an important place in the DTP environment. Users can output to their Postscript printer in the office and send the DTP files to a printing service bureau that supports Postscript output and get 1500 (or more) DPI output. However, this need belongs in the high-end solution and will be discussed in the next section.

The mid-range environment demands more sophisticated software products. DTP software such as Pagemaker and Ventura address even the most demanding DTP requirements. However, As you become comfortable with all the tools (both hardware and software) in the entry-level system, it is time to add software that addresses enhanced graphic and font needs.

Font packages are the fastest growing segment of the DTP software market. Many companies offer type face libraries that offer virtually unlimited styles and sizes of fonts. For example, products from Bitstream Incorporated offer a total of over 200 different fonts in 1990. Font vendors supply the DTP user with unique collections of typefaces. These typefaces may include script, small capitals, fancy script, shadow characters, fractions and ornaments.

Graphics software that allow freehand drawing and bit by bit manipulation allow the advanced user to create very exciting output to incorporate into your DTP software. Software such as PC Paintbrush IV allow you to edit scanned images bit by bit. This would allow a user to actually "correct" a flaw that was in the original of a graphic image or edit out something within the scanned image. Object oriented draw programs such as Corel Draw 1.1 allow for superb freehand drawing and incorporation of over 4000 different fonts from all the leading font vendors including Bitstream, Adobe and Compugraphic. In addition, Corel supplies the DTP user with a library of over 300 clipart images and has the ability to integrate over 10,000 images from 12 different clipart software vendors.

With the knowledge of what a faster processor and printer can provide and a better arsenal of advanced software, let's look at a mid-range DTP solution using a lower cost 16MHz 80386SX processor and a faster, 20MHz 80386 processor:

HARDWARE: (an "*" indicates a new or upgraded product over the entry-level configuration)	**US LIST PRICE AS OF 3/90**	
<u>80386SX SOLUTION:</u>		
HEWLETT-PACKARD QS16S 80386SX, 16 MHz clock speed 1-Megabyte RAM memory 5.25 inch floppy disk 84 megabyte hard disk VGA adapter One serial, one parallel port 7 I/O slots Desktop design	$4,699.00	*
Memory enhancement: 2MB expanded/extended memory	1,295.00	*
(QS16S allows a three MB configuration)		
-or-		

80386 SOLUTION:

HEWLETT-PACKARD QS20 80386, 20 MHz clock speed 1-Megabyte RAM memory 5.25 inch floppy disk 84 megabyte hard disk VGA adapter One serial, one parallel port 7 I/O slots Desktop design	$6,299.00	*
Memory enhancement: 3MB expanded/extended memory (QS/20 requires a one or four MB configuration)	1,947.00	*
MS-DOS 3.3	135.00	
VGA color display	749.00	*
Mouse	155.00	
Parallel cable	55.00	
Hewlett-Packard Laserjet III	2,395.00	*
Laserjet III memory: 2 megabyte memory upgrade	990.00	*
Hewlett-Packard Scanjet Plus scanner with interface kit	2,190.00	

SOFTWARE:	**US LIST PRICE AS OF 3/90**	
PC Environment: Microsoft Windows 2.11	149.00	
Word Processing Software: Wordperfect Version 5.0	495.00	
Graphics Software: Corel Draw 1.1	595.00	*
Desktop Publishing Software: Aldus Pagemaker Version 3.0	795.00	
-or- Xerox Ventura Publisher, 2.0	895.00	
OCR Text Scanning Software: ReadRight	495.00	
Font Management/Font Software: Bitstream Fontware Scalable Typefaces (includes five different font packages)	995.00	*

MID-RANGE CONFIGURATION CONTINUED...

Paint Software:		
PC Paintbrush IV	99.00	*

80386SX, Pagemaker Solution: $16,286.00
80386SX, Ventura Solution: $16,386.00
80386, Pagemaker Solution: $18,538.00
80386, Ventura Solution: $18,638.00

CONFIGURATION SUMMARY:
This configuration features the powerful 386 personal computer (16 MHz 80386SX or 20 MHz 80386 based processor) and will allow a user to create text, paint and graphics files with industry standard software and integrate these files into a full featured DTP software package. The output will be produced by what many industry sources feel is the defacto industry standard laser printer rated at eight pages per minute. With the additional printer memory, the user will be able to print a full page of text or graphics at 300X300 DPI with resolution enhancement and download an almost unlimited number of non-scalable fonts, depending on point size. The number of fonts is limited only to available printer memory since the larger the point size, the more memory the font occupies in the laser printer. In addition, the user will have access to all the scalable fonts resident in the laser printer. The scanner will allow the user to integrate graphic images or line art into the DTP software and read good quality printed material into a wordprocessor with the OCR software for integration into a DTP document. Finally, the paint software will allow image editing of scanned images and the graphics software selected will allow the user to integrate freehand art work into the DTP software.

Although there is a significant difference in price between an entry-level and mid-range system, the performance gain of the 80386 or 80386SX and the 8 page per minute printer will significantly improve a users' productivity. In addition, the specified software will provide many capabilities not typically found or needed in an entry-level system.

THE HIGH-END SOLUTION

When the objective of a company's DTP installation calls for the highest quality output for publications such as trade journals, publications for clients, annual reports, price lists or full color catalogs for customers, the hardware must be the finest and best performing technology. In addition, the software must allow for the highest quality output and uncompromising capabilities.

In 1990, the availability of 80386 machines running at 25 MHz or 33 MHz address the needs and performance expectations of a high-end DTP user. Although 1990 brought the industry the first 80486 class machine, capable of running 50 percent faster than a 33 MHz 80286, the performance improvements gained do not always justify their higher price. A user would be better to place their DTP budget dollars into additional printing capabilities such as fonts, clipart software or a Postscript solution.

Another productivity tool for the high-end user is a full-page or two-page monitor. These monitors provide super crisp displays that allow the user to view an entire DTP page or two DTP pages on the display. Full-page or two-page monitors are available in the following configurations:

FULL PAGE OR TWO-PAGE MONITORS

DISPLAY TYPE	SAMPLE ENTRY LEVEL PRICE RANGE	SAMPLE HIGH END PRICE RANGE
Monochrome	$349.00 (NEC Multisync GS)	$2,200.00 (Wyse WY-700)
Gray-Scale	$455.00 (Princeton Labs MultiView)	$3,500.00 (E-Machines Big Picture)
Color	$799.00 (NEC Multisync 2A)	$7,599.00 (Mitsubishi XC 3710C)

The above prices do not include a special adapter board needed to drive the high resolution, full page monitors. These adapter cards vary in price from $595.00 to $3,499.00. The prices of these monitors vary depending on resolution, one or two page display capabilities, screen size, scan frequency, and compatibility with current standards such as VGA and EGA. For illustration purposes, the high-end configuration used in the following paragraphs includes the gray-scale monitor from Princeton Labs. This competitively priced monitor provides the user with the ability to display an entire page of Pagemaker or Ventura (or virtually any MS-DOS application) at 800x1000 resolution with 16 shades of gray versus 640x480 with a standard VGA monitor. Princeton also provides a low cost adapter board ($595.00) and all the drivers needed to run sophisticated DTP software.

The final hardware enhancement for the high-end solution is the capability to output to a Postscript compatible printer. Because of the need at this level to create professional documents at the lowest cost (versus contracting outside services), the ability to print your documents on a Postscript equipped printer in-house at 300x300 DPI is vital. When the final copy or output is approved, the same DTP files can be processed by a professional printing house that has 1500 DPI (or more) Postscript capabilities for "magazine" quality.

Aldus Pagemaker and Ventura Publisher can compete successfully in all performance categories and can satisfy a user with the most demanding layout and page-makeup needs. There are some software packages that address only the high-end market. Packages such as PC News Layout from Information Engineering ($15,000.00), or Framemaker from Frame Technology Corporation ($2,500.00) or Pubset from Edco Services ($3,500.00) provide immense power and capabilities. These software packages typically provide an exclusive "DTP environment" that emulates word processing, graphics and page-formatting software.

As a general guideline, choose Pagemaker or Ventura when you plan to use your personal computer for other tasks such as spreadsheets, word processing, database applications or data communications. DTP users must be dedicated to DTP tasks in order to be successful with high-end DTP software "environments". These individuals must make a total commitment to learning and understanding the publishing trade and be able to transfer that knowledge to the PC environment.

The following represents a DTP system that will provide the user with state-of-the-art performance:

HARDWARE: (an "*" indicates a new or upgraded product over the mid-range configuration)	**US LIST PRICE AS OF 3/90**	
HEWLETT-PACKARD RS/25C 80386, 25 MHz clock speed 4-Megabyte RAM memory Cached memory architecture 5.25 inch floppy disk 155 megabyte hard disk VGA adapter One serial, one parallel port 7 I/O slots Tower design	$11,899.00	*
MS-DOS 3.3	135.00	
Princeton Publishing Labs MultiView Monitor	455.00	*
Princeton Publishing Labs GraphixPro Adapter	595.00	*
Mouse	155.00	
Parallel cable	55.00	
Hewlett-Packard Laserjet III	2,395.00	
Laserjet III memory: 4 megabyte memory upgrade	1,980.00	*
Adobe Postscript Cartridge	695.00	*
Hewlett-Packard Scanjet Plus scanner with interface kit	2,190.00	

SOFTWARE:	**US LIST PRICE AS OF 3/90**
PC Environment: Microsoft Windows 2.11	149.00
Word Processing Software: Wordperfect Version 5.1	495.00
Graphics Software: Corel Draw 1.1	595.00

HIGH-END CONFIGURATION CONTINUED...

Desktop Publishing Software:	
Aldus Pagemaker Version 3.0	795.00
-or-	
Xerox Ventura Publisher, 2.0	895.00
OCR Text Scanning Software:	
ReadRight	495.00
Font Management/Font Software:	
Bitstream Fontware Scalable Typefaces (includes five different font packages)	995.00
Paint Software:	
PC Paintbrush IV	99.00

Pagemaker Solution:	**$24,177.00**
Ventura Solution:	**$24,277.00**

CONFIGURATION SUMMARY:

This configuration features the powerful 80386 personal computer running at 25 MHz. This processor will allow a user to create text, paint, freehand, and graphics files with industry standard software and integrate these files into a full featured DTP software package. The software will provide top performance with the four megabyte system and cached memory architecture that the 80386 Vectra provides. In addition to a top performing PC, user productivity will be enhanced by a full-page, 16 shades of gray monitor configured with a 800x1000 high-resolution adapter card. The output will be produced on a Laserjet III equipped with four megabytes of memory and full Adobe Postscript capability. The user will be able to print a full page of text or graphics at 300X300 dots per inch using the Postscript language. The scanner will allow the user to integrate graphic images or line art into the DTP software and read good quality printed material into a wordprocessor with the OCR software for integration into a DTP document. Finally, the paint software will allow editing of scanned images and the graphics software selected will allow the user to integrate freehand art work into the DTP software. By adding the Postscript solution, the user will be able to send their DTP file(s) to an outside printing service for the highest quality, final document.

III. PRODUCTS THAT ENHANCE THE DESKTOP PUBLISHING ENVIRONMENT

The DTP user in 1990 can choose between what seems like an infinite number of products that enhance the DTP environment. For example, the 1989 buyers guide that was published in the September edition of Publish magazine listed over 3400 products for the DTP user. As an indication of growth for DTP products, the 1988 list published in the same magazine listed just over 1700 products.

Desktop publishing components can be listed by the following categories:

o...Word processing software
o...Page makeup software
o...Fonts - disk based and cartridge based
o...Paint and draw software
o...Graphic software
o...Operating environments (i.e. Microsoft Windows)
o...Scanning hardware and software including OCR software
o...Full page monitors
o...Personal computer hardware platforms
o...Printing hardware
o...DTP accessories (i.e. books)

As mentioned in the introduction of this paper, one component of a successful DTP installation is learning the techniques of your DTP software and perhaps more importantly, how to design an attractive, readable and effective document. The following books represents an excellent source of helpful information and will assist the DTP user to get the most out of their DTP hardware and software:

ALDUS PAGEMAKER:

Using Pagemaker, IBM Version
(Second Edition)
by S. Venit and Diane Burns; Que Corporation

XEROX VENTURA PUBLISHER:

Using Ventura Publisher
by Burns, Venit and Mercer; Que Corporation

LEARNING DESIGN TECHNIQUES:

Looking Good In Print
by Roger Parker; Ventana Press

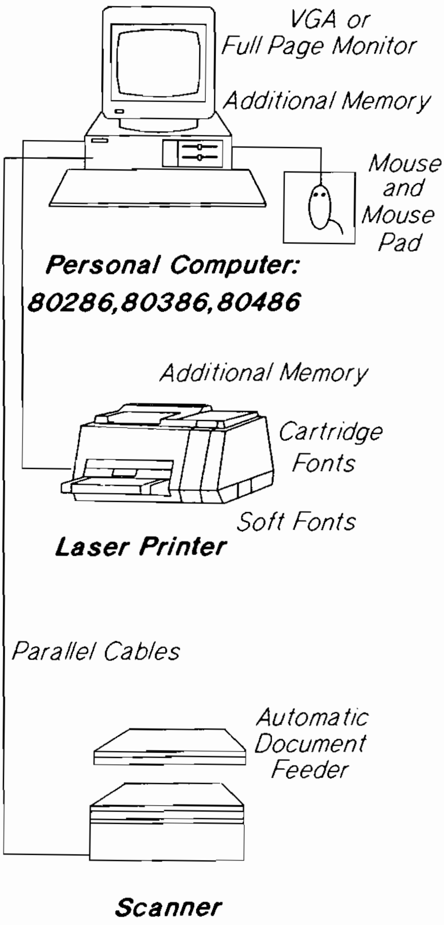
RECOMMENDED MAGAZINE:

Publish!; PCW Communications Inc., monthly

The illustration on the following page graphically represents how these components work together to produce a desktop publishing workstation:

PRODUCTS THAT ENHANCE THE DTP ENVIRONMENT

HARDWARE:



SOFTWARE:

- Operating System
- Operating Environment
- Wordprocessing
- Graphics, Paint & Draw
- Page Layout
- Spreadsheet

- Font Generation Software
- Disk Based Fonts (Soft Fonts)
- Font Mgt. Software

- Scanner Interface Software
- Optical Character Recognition

IV. THE 1990's: A LOOK INTO THE FUTURE

The users of desktop publishing components will enjoy many enhancements as new technology emerges. The 1990's began with the introduction of many exciting products such as the Intel 80486 processor, the new Laserjet III and PCL Level 5 from Hewlett-Packard, 300 DPI color printers under $10,000 and new DTP software enhancements. Let's conclude by taking a look at where the technology is heading as we enter the 1990's.

The high-end solution previously described specified a 25MHz, 80386 processor personal computer. A DTP user can be confident that this processor will address even the most demanding DTP performance requirements. However, as we enter the 1990's, computer vendors will begin to introduce and offer personal computers with the Intel 80486 processor. Intel says that the 80486, introduced at 25MHz during the early months of 1990, will run between two and four times faster than the 25MHz 80386 processor. The 80486 is a fast, 32 bit processor that has 1.16 million transistors and will maintain complete compatibility with all software that runs on the 80386. Intel plans to announce a 33MHz version later in 1990 that will provide even more impressive performance. The initial uses of an 80486 machine will be for a local area network server and high-end, single-user software such as CAD or DTP. Currently, 80486 systems are very expensive. An 80486 system can easily exceed $20,000. 80486 systems will drop in price as the price of the actual chip drops in price (currently, the 80486 chip alone has a list price of $995.00) and more vendors announce availability of their 80486 systems. When this happens, high-end DTP users must make a serious inquiry into what this state-of-the-art product can contribute to their productivity.

Two defacto standards have emerged for printer products of the 1980's. Printer Command Language (PCL) and Postscript will continue to be enhanced and will provide the DTP user with additional flexibility, capability and better performance at a lower price. New printers that use these printer language standards will continue to be introduced at a lower cost and better performance specifications as we venture into the 1990's. For example, the 1990 introduction of the Hewlett-Packard Laserjet III with a new version of the PCL standard introduced scalable typefaces that are contained within the printer and 600 DPI emulation at a cost savings of $300.00 over the Laserjet II printer.

The laser printer with PCL and Postscript has set the standard for low-cost, in-house printing devices. In fact, it was the introduction and vast proliferation of the low cost laser printer that allowed the DTP market to blossom. The 1990's will introduce many new exciting products. For example, 300x300 DPI color output, although available before 1990 will be an emerging technology that vendors will try to address at a price point that competes with low-cost desktop laser printers. Color technology will not be limited to laser devices. Low cost 300x300 DPI color ink-jet and thermal printers are certain to appear in the near future.

Software will continue to evolve and will offer DTP uses advanced features that will exploit evolving graphical interfaces. The two most popular and full-featured DTP software products have recently announced new versions of their software. Pagemaker, Ventura and other DTP software will continue to provide the DTP community with the tools that allow the DTP promise to be realized. One area of growth for DTP software will involve enhancements that will allow you to prepare your work for color publication in a cost effective and time saving manner. Ventura 2.0 for example allows the DTP user to apply six different colors plus black and white to DTP documents. You can preview your color output today on expensive, 300x300 DPI color Postscript printers or lower quality, 180x180 DPI ink-jet printers. As new technology emerges that will allow more DTP users to afford high quality, 300x300 DPI color printers, color printing will expand as new features are added to future software versions.

Ultimately, as new technology emerges, DTP users will be confronted with many decisions regarding industry standard printing languages, DTP software, operating environments, operating systems, personal computer platforms and other components that provide a DTP user with a state-of-the-art publishing environment. New technology will certainly provide exciting products, all designed to make the DTP user better equipped to produce camera-ready output. The DTP user will be the ultimate winner as the hardware and software vendors offer new technology that provide increased performance, lower price, more flexibility and new capabilities.

APPENDIX 1: ENTRY LEVEL DTP SOFTWARE

Affordable, easy to use DTP software is available for the casual DTP individual. These software packages, costing less than $300 do not provide the precise typographical control like a high-end package. However, the following products all support Postscript and PCL printers and provide a shorter list of features, making the software easier to learn and manage. These programs all support text import from ASCII, Wordperfect, MSWord and Wordstar and allow for column width adjustments. These products will provide you with acceptable results if your DTP needs are short and simple.

	MAX. PAGES	MANUAL KERNING	DRAWING TOOLS	ON-LINE HELP	AUTO TEXT FLOW
PUBLISH IT 1.12 Timeworks, Inc. Deerfield, Illinois 312-848-9200 **$199.95**	999	YES	YES	YES	NO
BYLINE 1.0 Ashton-Tate Torrance, California 800-437-4329 **$295.00**	99	YES	NO	YES	YES
GEM DESKTOP PUBLISHER 2.0 Digital Research, Inc. Monterey, California 408-649-3896 **$299.00**	100	YES	YES	NO	YES
FINESSE 3.0 Logitech, Inc. Fremont, California 415-795-8500 **$199.00**	16	YES	NO	YES	NO
PFS: FIRST PUBLISHER 2.1 Software Publishing Corp. Mountain View, California 415-962-8910 **$129.00**	99	NO	YES	YES	YES
SPRINGBOARD PUBLISHER 1.0 Springboard Software Minneapolis, Minnesota 612-944-3915 **$129.00**	999	YES	YES	NO	YES

APPENDIX 2: TRADEMARK ACKNOWLEDGMENTS

Ashton-Tate(R) is a U.S. registered trademark of Ashton-Tate Corp.

Microsoft(R) is a U.S. registered trademark of Microsoft Corp.

MS-DOS(R) is a U.S. registered trademark of Microsoft Corp.

Pagemaker(R) is a U.S. registered trademark of Aldus Corp.

Wordstar(R) is a U.S. registered trademark of Micropro International Corp.

IMPORTANT NOTE: Many of the software and hardware products that appear in this paper are marketed by various vendors under registered trademarks. These trademarks imply ownership of these products and the names of these products. This paper has placed registered trademark symbols only on products as required by specific HP software licenses and other agreements and has not placed trademark symbols next to all products that are registered and subject to state and federal laws. In no way should the reader imply that products with no registered trademark symbol are not registered trademarks. Please contact the vendor directly to confirm ownership of any U.S. registered trademarks.

Hewlett-Packard disclaims any and all liability for and makes no warranties, expressed or implied, with respect to the furnishing, performance, pricing or use of any product described in this paper titled "Desktop Publishing In The 1990's: What's New and What's Next" written by David Swinkin.

Distribution by Hewlett-Packard of the products or information concerning these products does not constitute endorsement of the product or supplier. Please contact an authorized Hewlett-Packard dealer for current pricing of Hewlett-Packard equipment. All Hewlett-Packard pricing is subject to change.

A Developers Perspective of NewWave

Scott F. Marshall
Innovative Information Systems, Inc.
63 Nahatan Street
Norwood, Massachusetts 02062
(617) 769-7511

No doubt most HP users have heard the fanfare or read at least one news release from Hewlett Packard about it's NewWave product. Recently, I've had the opportunity to develop a system using NewWave. This article will explain the current features and requirements of the NewWave office, and some of my experiences during the development effort.

I will not discuss programming for NewWave, but the issues facing a systems integrator. This article will start by explaining the hardware and software requirements. Next I will explain the major architectural components of the NewWave development environment. Finally, I will describe the effort necessary to integrate existing and new applications into the NewWave Office.

Hardware and Software Requirements

Hardware

286 AT 100% compatible
(Vectra ES 12)

4 Megabytes RAM

20 Megabyte disk drive

1.2 Megabyte high density drive

EGA resolution monitor and adapter

Mouse

These are the minimum requirements suggested by HP. I found that the 4 Megabytes of RAM were adequate, but if you plan to activate multiple applications, more memory is a necessity. The same goes for the disk space. The NewWave development environment requires approximately 6 megabytes of disk space to install. After adding the applications you plan to use, the 20 megabytes turns out to be a very tight fit. For example, Samna has written a NewWave version of Ami Pro that uses nearly 6 megabytes of disk space. Furthermore, as you integrate applications and create new data files NewWave creates a multitude of its own byte hungry files. A PC should have a 40 megabyte drive to run the NewWave development environment comfortably. Lastly, the system should include a pointing device. Like any other Windows applications NewWave is far easier to run and negotiate with a mouse. Like many longtime keyboard jocks I have never been convinced that the mouse could take the place of a keyboard. After a few weeks of using the mouse I've been proved wrong. A mouse is not just a substitute for typing but makes it far easier to organize and manipulate your work.

Software

DOS 3.2 or higher

Microsoft Windows 286 v2.11

Microsoft Windows Dev. Kit v2.0

Microsoft "C" Compiler v5.1 or higher

Microsoft Assembler (optional)

HP NewWave Development Kit

The only note about the software listed above is Windows 286 v2.11. Given the instability of Windows 386, HP has only released a development kit for the 286 version. At this writing, Microsoft plans to release Windows v3.0 in late May 1990. HP has plans to release the NewWave upgrade for Windows v3.0 shortly after the Windows release. The new version of Windows will be a single software package for both the 286 and 386 machines.

Environment

NewWave is essentially a user interface. Microsoft Windows, which NewWave runs under, is a user interface that separates the user from the DOS prompt. NewWave uses the Windows interface but goes a step further by separating the user from the DOS file system and devise control. All file listings and most commands have been replaced by icons that represent NewWave objects and tools.

The NewWave Office type interface is essential if computers are going to play a significantly greater role in the office environment. Personal computers have proven to be indispensable for everyday tasks such as word processing and managing spreadsheets. The next step is to integrate the applications and computers. There are integrated software packages, but the user is stuck with the options provided by the single vendor. We all know that different vendors have different strengths. NewWave allows for a mix and match of applications while retaining integration. The interface is also an industry standard. Learning overhead is very low since each NewWave application shares the same user interface.

Architectural Components

There are two architectural components of the NewWave Office that set it apart from the Windows interface alone. First is the Object Management Facility (OMF); the OMF tracks the status, composition and relationship of objects within the NewWave office. Second is the Application Program Interface (API); the API provides an interface between NewWave applications and system wide services. NewWave system wide services include the Agent (a system wide macro), Computer Based Training (CBT) and Context Sensitive Help.

Object Management Facility

The NewWave Office provides users with an object oriented environment. The idea behind the object interface is to insulate the user from any interaction with the PC's DOS environment. A NewWave office can contain four types of objects that replace the services and files of DOS.

User Objects. An application and the associated data file(s). These object are the simplest in the newWave Office and are created by the user.

Compound Objects. A user object that contains the representation of another user object. For example, a word processor document may have a chart within the text. For most applications, this would mean that the data file of the word processing application would contain the graphical representation of the chart. This is not so in the NewWave Environment. The chart within the text is really a "view" of the chart from another object. The word processor's data file only contains the text of the document.

Container Objects. One container object provided by NewWave is a file folder. To organize the NewWave office, User and Compound objects can be dragged into folders. Each folder has its own title (ie. "'90 Lotus Rpts") and may be dragged into other folders.

Office Tools. Computer services and utilities are represented by three dimensional icons called tools. The NewWave office includes a printer, waste basket, file drawer, diagnostic and others. To print, dispose, or store an object the user only needs to drag and drop the object onto a tool icon.

The OMF performs four primary activities to manage NewWave office objects.

- Application / Data Binding.
- Information Links
- Object Integration
- Object Communication

A user object is defined and maintained by the Application / data binding functions. Information links, object integration and the different object types, are illustrated in figure 1.

Figure 1 represents the primary information links provided by the OMF; simple, visual and data. In figure 1 the spreadsheet and chart relationship illustrates a data link. When spreadsheet data changes, the OMF will note the change, and the next time the chart is viewed it will reflect those changes. The chart is visually linked to a document, so the change in the spreadsheet also updates the document. All the updates are recorded and controlled by the OMF.

The OMF supports communication between objects using the Windows messaging system and enhancements offered by NewWave. The Windows messaging system controls all communication between the user (keyboard and mouse) and the application. Windows also controls the application's output to the screen. All communication within the Windows environment is between the user and an active application, or between active applications. NewWave enhancements support communication between active and inactive applications. This is the functionality needed to support the dynamic links between objects.

Application Program Interface

The principal component of the API is the Agent. The Agent is a NewWave system wide service that will automate user tasks. Unlike the macro languages associated with programs like Lotus 1-2-3, the Agent is capable of working between all applications within the NewWave office.

If a program has not been specifically written for NewWave the Agent may communicate with an application using a keyboard macro. If the application is a NewWave program, and written with API support, the agent interfaces with the application using NewWave's messaging system.

The Agent differs from other marco languages because it does not record the actions necessary to complete a task.

Instead, the agent records the result of an action. For example, A typical macro language will record the movements of objects in the window. The movements recorded are absolute. If the macro is played back and the object is not in the same location on the screen, the object's destination is also different. What if the user originally records the dragging of an object to the printer? The playback would only work if the object to print and the printer object are in the same position as originally recorded. NewWave Agent does not record the object movement, but records the command to print the object.

The API also gives the application programmer access to the Computer Based Training (CBT) and Context Sensitive Help system services. The CBT is closely related to the Agent. A CBT system is an agent task that monitors and guilds the user though the application. Context Sensitive Help services allow the programmer to include "hooks" in the program that respond to a request for help. The response identifies the context and the appropriate help text to display. The programmer does not have to be concerned with the actual text or its format.

Application Integration

The ability of an application to take advantage of the OMF and API depends on the level of the program's integration into the NewWave Office. HP has defined four levels of application integration.

1) DOS or Windows application executed within NewWave

2) Encapsulated application

3) Advanced encapsulated application

4) NewWave application

The first three levels are intermediary steps. HP considers tools to encapsulate existing programs into the NewWave Office as a temporary fix. HP is expecting major vendors to rewrite their software to run within the NewWave Office.

DOS or Windows Applications

A DOS or Windows application executed within NewWave is the lowest integration level. An application at this level cannot take advantage of any OMF or API functions. It takes a user or developer approximately 5 minutes to include an application in the NewWave Office at this level. NewWave prompts the user for the program name, path and program runtime information to store in a Program Information File (PIF). The PIF includes information about a program's memory requirements and screen attributes. The benefits of this level of integration is that it is quick, easy and allows the user to execute nearly any program on the system without leaving NewWave. Only DOS applications with large data files were unable to be integrated into the NewWave Office.

Encapsulated Applications

An encapsulated application incorporates some OMF facilities. The application is "bridged" to the NewWave Office that gives the application object like characteristics. The application can be represented as an object in the NewWave Office and manipulated like any other object. The encapsulated application can be bound to data files, opened, closed, moved, copied, filed, discarded and mailed. The encapsulated objects may not be printed outside the application and does not support information links. The encapsulated application does not support any of the API services either.

The encapsulation process takes from 20 minutes to two hours, depending on the options the user or developer includes. Most of the process is prompted through menu screens. The process includes selecting an icon to represent the application, which file extensions can be used for data files, templates and custom menus. After some practice, the basic process of encapsulation is very simple. However, it's not possible for the typical user to incorporate custom menus and other advanced options.

I found that the encapsulated objects behavior could be inconsistent. One of the applications I encapsulated consisted of three individual Windows programs. The first program ran from an agent task that sent the application a keystroke macro. The second program would return an error if I tried to record the keystrokes or play back a keyboard macro I created myself.

Advances Encapsulation

Advanced encapsulation involves the same steps as encapsulation but also involves the development of a new application. The new application is called a "browser" and allows the application to take advantage of the OMF information links. It requires a significant programing effort to create a browser. HP has distributed the encapsulation and browser for Lotus 1-2-3 and Gallery (HP's own graphics package) with NewWave. The browser allows the application to share its data with NewWave applications. In figure 1 the chart "view" within the word processing document is possible because the charting application is supported by a browser.

Advanced encapsulation does not support information links that "share-in" data. Using the previous example, this means that the word processing object cannot be viewed within the chart. The ability to share-in data and support the API services are limited to NewWave applications.

NewWave Application

A NewWave application is a program specifically written to take advantage of the Windows' user interface and NewWave's OMF and API services. Applications that currently run with NewWave are from vendors whose applications were previously written for windows. The modifications necessary to have a Windows program run with NewWave is far less time consuming then converting a DOS program. NewWave applications being released at this time typically include OMF support, but not all the API system services. This is because it is a far more involved process to use the API system services then the OMF capabilities.

As of this writing there are only a handful of NewWave applications distributed by vendors other then HP. Below is a list of the applications I have evaluated.

DynaComm Asynchronous / Terminal Emulation
Future Soft Engineering

Ami Pro / Word Processing Samna (Beta version)

Desktop Accessories
Dexoteck

This is not a very long list. But it is expected that most developers will release new Windows and NewWave applications when Microsoft releases Windows v3.0.

I will also caution users not to expect too much from the API in the short run. Most application developers will continue to use their own help and macro facilities as they port to NewWave. The functionality of the Agent within an application also depends on the programmer. NewWave Office, without question, is a revolutionary PC environment. If enough application developers write for the NewWave Office, it couldn't help but be a hit with users.

OBJECT CLASSIFICATIONS / RELATIONSHIPS

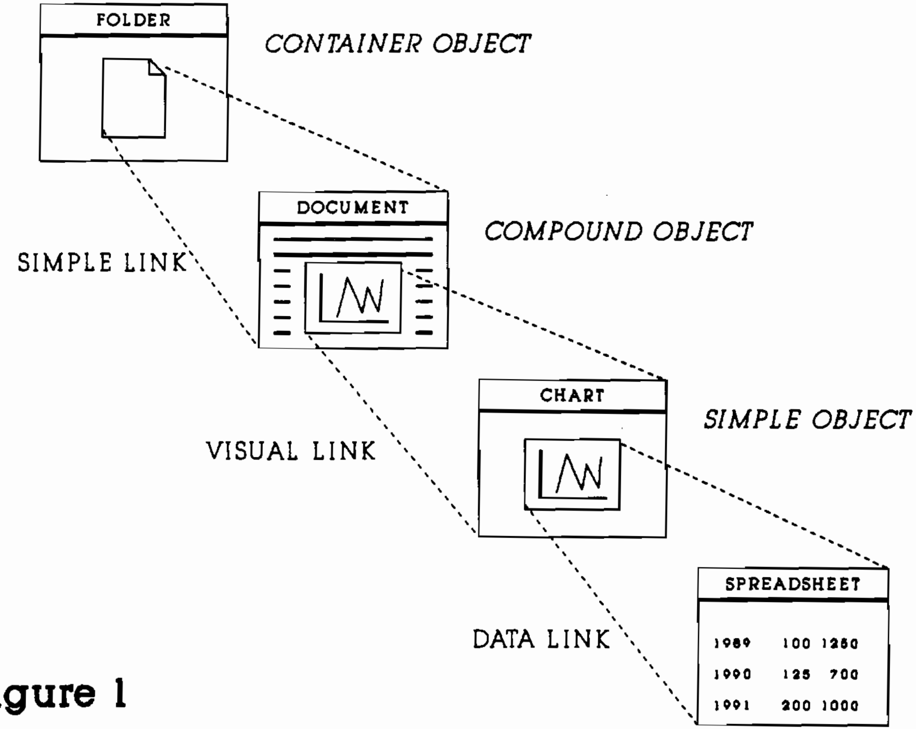


Figure 1

TITLE: A Client/Server Model

(Sales/Lead Tracking and Management)

AUTHOR: Bob Hoover

Infocentre Corporation

5674 Stoneridge Dr., Suite 218

Pleasanton, CA 94566

415-460-8220

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 4231

TITLE: Building A Software & System Support

AUTHOR: Don Jones

INLEX Inc.

PO Box 1349

1900 Garden Road, Suite #160

Montery, CA 93942

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 5023

Case Studies of MPE/XL Migrations

By Paul S. Thomas and Brent M. Ehrman

Crowe, Chizek and Company
10 West Market Street
Suite 2100
Indianapolis, IN 46204-2976
(317) 632-8989

Over the last two years, we have had the opportunity to assist many companies in their migration from 'Classic' to 'RISC' technology. As one would expect, almost all cases of migration to MPE/XL technology are precipitated by the need for additional system performance. Rarely would a company migrate from MPE to MPE/XL for any other reason. In each of the following cases, the companies were in fact searching for performance improvements from two to three times their current system performance.

Case # 1:

The first case involves a company whose entire primary application is written in the PROTOS 4th generation language. This application involves significant on-line and batch processing and requires optimal performance for daily completion of processing. The application predominately uses TurboIMAGE file structures.

The application involves on-line order entry with over 40 order entry clerks. These clerks are entering orders on-line while talking to customers over the phone and, during slow periods, are involved in entry of orders sent in via mail. Many customers call in to check on order status and to request information about available products.

Additionally, there is on-line usage by warehouse personnel for inventory management, order fullfillment, and invoicing. In total, there are over 120 terminals connected to this system.

The batch processing requirements for this system consist of nightly processes that receive orders electronically via EDI, process order history, and perform order maintenance processing.

The company has significant processing volume. The company receives and ships over 3,000 orders per day with the average order containing four line items. The item master file contains over 2 million different item master records. The demand for the items not stocked in inventory are sent to a manufacturing system on another HP/3000. Once manufacturing has completed the items, information is transmitted back to the order entry system and pick tickets are printed for shipment to customers.

The 'Classic' system configuration consisted of an HP/3000 Series 70 with maximum main memory and with 24 Eagle disc drives (13 Gigabytes). Nightly batch processing required eight hours to complete. Nightly backup had only three hours to complete. Weekend processing, month-end processing and system backup was taking longer to complete than the 60 hours available during the weekend and was spilling over into Monday's on-line time.

The migration process had to occur in such a manner as to not disturb the normal processing of orders. A new HP/3000 Series 950 was ordered complete with new disc drives. The migration process and testing occured on the new system using a copy of the software and data bases from the existing 'Classic' machine.

Several issues and considerations in the migration required investigation prior to cut over. The most basic of these was the question of how to decide which programs to migrate to native mode versus continuing with use of compatibility mode.

Our first reaction was to move all of the software and data to the new machine in compatibility mode then over time migrate programs to native mode. We decided to run a benchmark test of the monthly batch processing in compatibility mode to determine the performance gain of the HP 950. To our surprise, the monthly batch processing in compatiblity mode was slower on the HP 950 than on the Series 70. As a result, we converted the month-end batch programs to native mode and ran a test again. This time, we obtained a performance gain of 364 percent which exceeded our expectations.

Since our basic need for improved performance was in the nightly, weekend, and month-end processing, we focused our efforts in migration to these batch programs. The on-line programs would remain in compatibility mode on the new machine and would be converted later. This allowed us to accelerate the migration process to the new machine and quickly receive significant performance benefits.

To perform the software migration to native mode, tools such as QEDIT and MPEX were invaluable in allowing mass software searches, changes, and recompiles. The actual time spent migrating the software to native mode was insignificant compared to the total planning and testing time. As with any conversion of software, the time spent testing a major software system greatly outweighs the actual code conversion effort.

The final results from the migration showed batch performance improvements ranging from 60% to over 400%. These improvements were in such a broad range that we spent time with HP performance specialists to understand the causes of the varied performance results. Our basic determination was that since we were utilizing such large data bases with hugh capacities, the type of program process that was utilizing the data would greatly impact performance. For instance, if the program was to serially or sequentially read data from the data bases, the data would be loaded into main memory in large blocks and would result in I/O time reduction thus resulting in performance improvements. But, if the program was somewhat random in the requests for I/O, since the data bases were so large in size, rarely would the requested data already be residing in main memory and this would require a physical I/O and result in lower performance improvements.

An example of a serialized process would be reading a data set to extract data for a report, sorting the data, and printing the report. This allows an 'XL' machine to bring in large blocks of data into main memory and improve program performance. An example of a more randomized process would be the processing of new order information where each new order would cause I/O to different physical data blocks within a data base. The program would probably be requesting data blocks that are residing on disc rather than in main memory thus requiring multiple disc I/O's and slowing performance.

CASE # 2:

This case involves a company whose applications are written primarily in COGNOS. The company has approximately 100 terminals using applications such as Order Entry, Job Costing, Cost Accounting, Job Tracking and Scheduling, Accounts Receivable, Accounts Payable, General Ledger, Purchasing, Shipping, and Inventory Management.

The 'Classic' system was an HP/3000 Series 70 with maximum main memory and 6 Eagle disc drives (3.4 Gigabytes). The applications use almost an even mix of KSAM and TurboIMAGE file structures. The migration was further complicated by the requirement to upgrade much of the COGNOS software from previous versions (versions 4.00, 5.01, and earlier) to COGNOS versions supported on the HP 950.

HP found it particularly difficult to measure and predict the performance improvement of applications written in 4th generation languages. HP performance tools could not effectively isolate the particular user programs executing and causing performance problems. HP also finds it difficult to assist in migration assistance when software and tools other than those supplied by HP are used. This caused us to conduct performance benchmarks to assess the impact of the migration.

During testing of COGNOS QTP programs migrated in compatibility mode on the HP 950, performance improvements of 20 percent were realized. A more significant performance improvement was needed.

It also became very difficult to compare performance between the 'Classic' machines and the 'XL' machines. The CPU seconds required for execution on 'XL' machines were significantly less than the CPU seconds required for 'Classic' machines. We attempted to develop a ratio of performance change by using the change in CPU seconds compared to the change in elapsed time, however, this did not consistently match. Although one would expect that less CPU seconds would mean less overall work for the CPU resulting in increased overall performance, we were cautious to expect significant overall gains since we did not acheive the respective increase in performance in elapsed time. The only effective method for performance measurement was to compare 'elapsed time' for processing.

The COGNOS software was written using 'SPL' on the 'Classic' machines and rewritten by COGNOS in 'C' for 'XL' machines. This rewrite by COGNOS caused many significant issues. Many features and functionality of COGNOS were changed to perform differently under the 'XL' environment. COGNOS changed some of the processing timing and ordering of instructions. As a result of these changes, many programs had to be changed and/or rewritten as part of the migration. Additionally, since COGNOS had rewritten their software, many new software bugs were discovered during the migration process. The 'real world' enviroment could not wait for vendors to fix their bugs, thus resulting in the rewriting of programs to work around the COGNOS software bugs.

As a result of migration and conversion of programs to native mode COGNOS on the HP 950, we were able to obtain the expected 100 to 200 percent performance improvements.

Since we incurred so many difficulties in the migration due to COGNOS compatiblity issues, we were required to throughly test all of the applications prior to running live. This indepth testing resulted in finding and fixing many problems and ensured a successful migration of the live users and data. If we would not have thoroughly tested the application software, we would surely have headed for disaster.

SUMMARY

Although MPE/XL offers many new features and additional functionality over 'Classic' MPE, almost all migrations to MPE/XL occur due to meet the system performance needs of a company.

As with most upgrades, a performance improvement of 100 percent is desired. Performance improvements greater than 100 percent usually leave company management with the desire to upgrade to a less powerful machine and upgrade again later if additional performance is needed. This is where Murphy's Laws come into play and tends to frustrate company management. The amount of required resources will rise to exceed the amount of available resources. This has proved valid time after time in our experiences with companies.

Within 12 to 18 months after an upgrade to a new machine that is supposed to last 3 to 5 years, performance problems begin to occur. The companies look back in shock and ask themselves, has our volume increased so significantly to require more resources? Have we added more new applications that require more resources? Have we really grown that much?

This is when HP is usually called in to justify why the system that HP sold them has failed to last as long as promised. Usually HP will send in performance consultants to wade through the volumes of performance data, produce beautiful color charts with peaks and valleys, provide a written report full of information, yet fail to provide management with the understanding that they have in fact changed. Management almost expects HP to perform some type of voodoo on the machine in hopes of reviving the lost system performance. Rarely can HP provide an answer other than they need more this or they need more that (memory, disc channels, or cpu).

Performance effects vary from application to application and from 'Classic' to 'XL'. It is not possible to make a general statement that if you are using a 'Classic' machine that you will experience performance improvements by moving to 'XL'. Rarely will someone experience a performance degradation by moving from 'Classic' to native mode 'XL', but the move to compatibility mode 'XL' may result in slower performance. It is also not possible nor resonable to make an across the board prediction as to the improved performance on 'XL'. As has been seen by the many performance studies conducted by HP and the war stories told by many HP users, post migration performance improvement varies from program to program.

Based on our experiences with migration to MPE/XL, we would suggest the following actions as part of your migration:

1. Test the performance changes to critical programs prior to migration.

2. Set management expectations correctly. (i.e. don't let them assume an across the board performance improvement of 100 percent).

3. Test application functionality thoroughly after migration and prior to running live (especially for 4th GL applications).

4. Initially migrate only the critical programs to native mode.

5. Plan the details of all aspects of your system migration in advance. There are so many details that if not properly planned, many will cause disastrous ramifications.

After all the testing and benchmarking of performance improvement, 'real world' results are always diluted by changes in the business environment.

Title......Controlling the Development Process

Author.....Thomas P. Eineker

Company....Paul Mole' Consulting, Inc. (PMC)

45 Brant Avenue

Clark, NJ 07066

(201)-381-5400

Software development projects fail for many reasons. Most of these reasons can be traced back to ineffective communications. A project approach that ensures effective communications between all project members, combined with the key elements of a strong "Project Management Process", a solid "Software Development Methodolgy" and an effective "Work Plan Approach" will greatly enhance the chance for a successful project. The following discussion details the structure to be used as a guide for understanding the key elements for effectively communicating and controlling the software development process:

1. The Five Stages in the System Life Cycle of the Development Process (Management overview)
2. The Six Most Common Problems with Development Process
3. An Outline for the "Project Management Process"
4. An Outline for the "Systems Development Methodology"
5. An Outline for a sample "Work Plan Approach"

All personnel involved in a software development project should clearly understand the role of each group during the project cycle. Three groups are usually represented as .....

1. Executive management normally participates in the project to insure the outcome is benefical of the organization. The members of this group perform the functions of approving expenditures, make high level business operational decisions, arbitrate in the event of a deadlock and continually review project status.

2. Project team members are those individuals who perform the detail tasks of the preject. Some functions this group performs are producing the required project deliverables, constantly adjust project status details and report project status to the executive committee.

3. Users are personnel who will eventually take advantage of the software being developed. The main functions performed by this group are informing the project team as to the requirements for the software, reviewing and signing off the documents detailing these requirements and final testing and signing off of the completed programs and documentation.

The communication between the members of these groups are both of a formal and informal nature. To insure that communication is effective, a couple of simple guidelines need to be observed......

1. communication regarding project details must be formal and structured
2. communication must occur regularly
3. communication must be accurate, objective and verifiable
4. all project communication should flow through a focal point

The project manager performs a direct role in project management. Along with the task that the project manager is directly responsible for, this individual also performs the role of salesman, coach, score keeper, cheerleader, and diplomat. Although these are items not found on a project task list, they are equally important.

The tasks which the project manager is directly responsible are reporting project status to executive committee, carry out executive committee decisions, assign and follow-up on tasks given to project team members, enforcing the software development methodology and project schedule, addressing problems and obstacles to the appropriate individual and escalating problems and obstacles to the appropriate level.

The most important of these task is reporting project status to the executive committee. Although the project manager contributes heavily to the project's success, this individual alone is not responsible for it. The project manager's main responsibility is to provide the executive committee with the information necessary to make effective decisions.

The informal roles the project manager plays are more related to creating an environment and attitude conducive to the project success. The project manager must make all persons involved with the project feel the project is of benefit to them personnally and that they play an important role. Along with this, the project manager must mend fences between project members and help cut off potential problems before they occur.

The relationship between the executive committee and the project manager is very important. The executive committee must have confidence in the project manager's ability to solve problems, carry out project related decisions and in the information supplied by the project manager. The executive committee must support the project manager and respond to the request made of the executive committee to exercise its authority.

First we need to understand the development process from a general point of view. The following shows the FIVE major stages in the life cycle of a software development process. The five(5) major steps are listed down the left side of the page followed by short definition or meaning.

```
***************
*REQUIREMENTS* Determining what the system is supposed to do.
*            *  Documenting the requirements completely.
***************


***************
*DESIGN       *   Creating a good overall system structure. Assign
*             *   tasks to modules. Defining module interfaces.
*             *   Checking that the design meets the system
*             *   requirements.
***************


***************
*PROGRAMMING * Coding and debugging the individual modules.
***************


***************
*INTEGRATION *  Putting the modules together. Testing the running
*            *  program against the system requirements.
***************


***************
*MAINTENANCE *   Correcting errors discovered by users. Adding new
*            *    functions requested by the users. Making system
*            *   changes required by changes in its interfaces to
*            *   other systems.
***************
```

There are some problem areas that we feel are important to watch closely because most projects fail due to these problem areas...

1. Definition/Scope understood and well thoughtout

2. System Design clearly stated

3. Programs coded with consistentcy/standards

4. Tracking/Reporting issues,problems,tests,bugs,enhancements..

5. Testing Methodology/Experience lacking

6. Producing/Enhancing documentation

Our company has addressed these problem areas by developing the following procedures

1. We've developed a "SYSTEM DEVELOPMENT METHODOLOGY" which has been extracted from the use of Case Tools, 4GLs, System Life Cycle, Yourdon "Structured Analysis", and our own library of knowledge.

2. We've developed own "PROGRAMMING STANDARDS MANUAL".

3. WE've developed a "TEST PROBLEM REPORTING/TRACKING SYSTEM".

4. We've produced a formal TESTING/REPORTING method.

5. We have successfully produced "WORK PLAN APPROACH" that combine the "SYSTEM DEVELOPMENT METHODOLGY" and the "PROJECT MANAGEMENT PROCESS".

Now we should review the basics for the "Project Management Process". This "project Management Process" can act as guide or checklist for the task you need to have your project manager review before your project runs into problems.

We have established a process for improving the communication, structure and direction of our software projects to produce a better quality system for our customers. The following details some of the basic principles. There are many other supporting forms, procedures, guidelines, checklist, plans, and standards that are used along with the enclosed information.

PROJECT MANAGEMENT PROCESS

PLANNING

- Define objectives
- List tasks
- Estimate effort and duration
- Determine the task interdependancies
- Schedule the tasks
- Schedule the resources
- Balance the resources
- Obtain approval of the plan

DIRECTING

- Review the plan objectives with the team
- Assign the tasks
- Review the criteria for each task completion
- Motivate the team

CONTROLLING

- Review progress
- Control the changes
- Report the progress
- Replan
- Conduct project reviews
- Review and approve completed work
- Assure delivery of project products
- Resolve issues
- Close project

ADMINISTERING

- Recruit
- Develop personnel
- Evaluate performance
- Develop policies, standards, guidelines, and procedures
- Maintain policies, standards, guidelines, and procedures
- Budget appropriately

WORK PLAN APPROACH - (for a sample company "ABC")

The requirements stated in the Request for Proposal were reviewed in light of proven computer system development practices and methodologies and our previous experience designing and implementing similar computer systems. Based on this review, we have prepared a work plan founded on sound computer development practices and tailored to meet the needs of this particular project. The work plan is divided into fifteen tasks summarized below.

TASK 1: INITIATE PROJECT

This task includes the steps necessary to get the project underway. During this task the project team will become familiar with the ABC organization and its operation and policies. In addition, the project work plan, schedule and deliverables presented in our proposal will be finalized and submitted for approval to the management of ABC. The following sub-tasks will be performed:

1.1 Initiate Project
1.2 Conduct Project Orientation
1.3 Finalize Project Scope
1.4 Review ABC Standards & Procedures
1.5 Install Project Committees
1.6 Arrange Project Facilities
1.7 Finalize Work Plan, Schedule & Deliverables

TASK 2: REVIEW EXTERNAL DESIGN

Work with ABC project committee to prepare a detailed specification of the system - the External Design. Task 2 will focus on the final design of the system. The following sub-tasks will be performed:

2.1 Prepare (with Committee) the Detailed External Design Document
2.1 Review External Design Document
2.2 Review Related Systems Documentation and Specifications
2.3 Identify/Resolve Issues Concerning External Design

TASK 3: REVIEW BASE SOFTWARE PACKAGE

After fully understanding the requirements we will compare the final requirements to the base packages. Those features included in the packages will be noted. Required modifications and additions to the packages will be identified, defined, categorized and priorities set. The following sub-tasks will be performed:

3.1 Compare External Design Specifications to Base Packages
3.2 Identify and Classify Necessary Modifications
3.3 Prioritize Modifications
3.4 Review Third Party Software

TASK 4: REVIEW SYSTEM ARCHITECTURE

During this task, the various architectural components from which the system will be built will be finalized. This task includes the final specification of system software and hardware configuration based on the system functions and data requirements decided on above. The following steps will be performed:

4.1 Finalize Characteristics of Each Subsystem
4.2 Finalize Logical Data Base Design
4.3 Finalize Hardware and System Software Compatibility
4.4 Review System Architecture with ABC

TASK 5: PREPARE IMPLEMENTATION PLAN

The purpose of this task is to anticipate the requirements for implementation well in advance of the actual implementation. Activities take into consideration equipment capacity, required system & utility software and the availability of operations manpower/clerical resources required for data conversion and system acceptance testing. This task charts a course for the remainder of the project. The steps to be performed are:

5.1 Determine the Sequence of Subsystem Implementation
5.2 Determine File Conversion Strategy
5.3 Develop Testing Strategy
5.4 Review Hardware/Software Installation Strategy
5.5 Review Implementation, Interfacing & Integration
5.6 Develop Detailed Implementation Work Plan & Schedule
5.7 Prepare Implementation Plan Document
5.8 Review Implementation Plan Document with ABC

TASK 6: PREPARE DETAILED SPECIFICATIONS FOR PACKAGE MODIFICATIONS

During this task, we will expand the external system design to fully define reports, screens, inputs, outputs and processes. Preliminary program specifications will be defined, the data dictionary will be expanded, the physical data will be modified and system security and control issues will be addressed. The steps to be performed include:

6.1 Prepare Report Specifications
6.2 Prepare Input/Output Screen Specifications
6.3 Refine Logical Data Base Design
6.4 Finalize Transaction Definitions
6.5 Finalize Input/Output Methods
6.6 Finalize System Control and Audit Requirements
6.7 Define Redundancy/Recovery/Restart Facilities
6.8 Prepare System Overview Diagram
6.9 Develop Change Control Procedures
6.10 Prepare Preliminary Test Plan Outline
6.11 Prepare Programming Schedule
6.12 Finalize Detailed Specification Document
6.13 Review Detailed Specification Document with ABC

TASK 7: DESIGN PROGRAMS

We will design and document the programs identified earlier in program specification packages. These packages will be used by the programmers to modify existing programs, customize, or code new programs as required. We will prepare a specification package for each program identified. The specification package will contain the information from which the program code will be written. Each program specification package will include:

A. Program Overview Narrative
B. Program Macro-Level Diagram
C. Program Structure Chart
D. Detailed Logic Charts - when needed
E. File/Data Base Specifications
F. Input/Output Layouts
G. Edit/Validation Criteria

TASK 8: CUSTOMIZE PACKAGES/CODE MODIFICATIONS

We will customize the base-line packages in accordance with the External Design document requirements. This customization will take into account the following:

A. Base-line code
B. Customization
The following sub-tasks will be performed:
8.1 Data Base Specifications
8.2 Screen Layouts
8.3 Transactions
8.4 Security
8.5 Edit/Validation of Field data

TASK 9: UNIT TEST MODIFICATIONS

During this task, we will develop test plans for each program and each program will be tested to ensure that it executes properly prior to performing integration and system testing. The steps which we will perform are:

9.1 Establish Unit Test Environment
9.2 Develop Unit Test Plans
9.3 Generate Unit Test Data
9.4 Execute Unit Tests

TASK 10: SYSTEM TEST MODIFICATIONS

During Task 10, we will combine and exercise the processing functions of the system to ensure that they integrate properly. Our approach to system testing is structured to ensure that all conditions are tested in a manner that facilitates the identification of problems and errors. Our tests are designed to ensure the intricacies of data interdependence. The steps which will be performed are:

10.1 Finalize System Test Strategy
10.2 Review System Test Strategy with ABC
10.3 Develop Detailed System Test Plan
10.4 Develop System Test Data
10.5 Establish System Test Team
10.6 Conduct System Test

TASK 11: PREPARE USER PROCEDURES AND SYSTEM DOCUMENTATION

This task will be performed concurrently with other tasks as the system is developed to ensure that the user procedures and documentation complement the system. The user procedure manual is tested along with the system during the system test task. Our approach includes the following key elements:

A. Document manual as well as automated procedures;
B. Prepare separate manuals for users, computer operators and system analysts/programmers;
C. Review and obtain approval of all documentation from ABC before distribution.

11.1 Prepare Functional Procedures
11.2 Prepare System Overview Documentation
11.3 Prepare Project Notebook
11.4 Prepare User Manuals
11.5 Review Procedures & User Documentation with ABC

TASK 12: TRAIN USERS

During this task, we will prepare participant course materials for training sessions. Preparation of training materials is done concurrently with the development and testing of the system. The steps which we will perform are:

12.1 Define Participant Groups & Training Needs
12.2 Prepare Course Materials for Participants
12.3 Conduct User Training by Participant Group

TASK 13: OBTAIN USER ACCEPTANCE AND IMPLEMENT

We will implement the system and conduct acceptance testing during the task. The steps we will perform are:

13.1 Finalize Acceptance Criteria
13.2 Prepare for Implementation
13.3 Conduct Data Conversion
13.4 Conduct Acceptance Testing
13.5 Develop Disaster Recovery Strategy Plan
13.6 Obtain ABC Formal Acceptance

TASK 14: CONDUCT POST - IMPLEMENTATION REVIEW

We will conduct a post-implementation review of the system. The purpose of the review is to ensure that the system is performing as expected after full implementation. The following sub-tasks will be conducted during post implementation review:

14.1 Evaluate System Information Flow & Operation
14.2 Evaluate User Satisfaction with System
14.3 Evaluate Accuracy & Timeliness of Processing
14.4 Prepare System Review Report

TASK 15: MANAGE THE PROJECT

This task ensures that the project objectives are understood by the project team and ABC. This task is ongoing throughout all phases of the project. Steps performed are:

15.1 Monitor Project Progress
15.2 Prepare Project Status Reports & Briefings
15.3 Conduct Project Meetings with ABC
15.4 Follow "Project Management Process"
15.5 Establish/maintain Project Book

SYSTEM DEVELOPMENT METHODOLOGY

A. Feasibility Study

B. Requirements Specification

C. Conceptual System Design

D. System Architecture

E. Implementation Planning

F. Detail Design

G. Program Design

H. Program Coding

I. Unit Testing

J. System Testing

K. Procedures and User Documentation

L. Training

M. Implementation and System Acceptance

N. Post Implementation Review

O. System Documentation

A. Feasibility Study

1. Review the existing system
2. Outline the required system
3. Evaluate the system alternatives
4. Develop time and cost estimates
5. Complete and submit a feasibility study document/report

B. Requirements Specification

1. Identify the structure and business of the organization
2. Identify the scope of the project
3. Establish and maintain "Project Book"
4. Prepare Questionnaires
5. Conduct interviews
6. Document and analyze data
7. Document and analyze work processes
8. Define information requirements and data attributes
9. Document proposed system description
10. Document the functional/information requirements
11. Transform the requirements into structured process specifications
12. Summarize and categorize benefits for requirements
13. Assess existing application systems
14. Select the development approach for the system (ie. tailored system development, software package, prototyping) with reasoning, associated costs/benefits
15. Complete and submit reqiurements specification document with management summary and recommendation

C. Conceptual System Design

1. Organize process specifications into logical groups
2. Establish an overall system design
3. Select a development approach which will be the rules, standards, guidelines, procedures to be followed by this team for this project.
4. Prepare system design specifications
5. Determine system control and auditing
6. Determine testing, conversion and implementation needs
7. Determine preliminary operations data

D. System Architecture

1. Development of the conceptual data model showing each users view of the data along minimum data relationships.
2. Define the functional characteristics of each subsystem
3. Assess the feasibility of distributing system processing
4. Prepare preliminary telecommunications requirements/costs
5. Define the file structure
6. Complete logical data base design
7. Summary of walk thru of logical data base design looking for functional problems, response problems, operational considerations...
8. Finalize system architecture document
   - Logical data base diagrams showing internal/external relationships
   - Screen and report contents
   - Functional processing flows
   - File structure recommendation
   - Logical data base design
   - Distributed processing strategy
   - Telecommunications network diagram

E. Implementation planning

1. Hardware configuration diagram/document
2. Educational staff plan
3. Technical personnel resources required
4. Clerical personnel resources required
5. Testing and Conversion plan
6. Develop an implementation checklist
7. Develop an implementation schedule
8. Estimate computer resources needed
9. System security plan

F. Detail Design

1. Input document layouts
2. Screen display layouts
3. Initial system diagram
4. Screen hierarchy diagram
5. Program macro diagrams
6. File definition layouts
7. Edit and validation specifications
8. Actual data set definitions and characteristics
9. Statement of processing requirements for program and file
10. Statement of recovery/restart provisions
11. Report specifications
12. Requirements cross reference
13. Special design considerations
14. System flowcharts

G. Program Design

1. Amend program input and output requirements
2. Finalize program processing requirements
3. Complete program logic charts
4 Finalize program design error messages,specifications...

H. Program Coding

1. Complete establishment of coding standards and conventions
   - Structured programming Standards
   - Naming conventions for programs, group names, modules, copylibs, member names, files, data sets, paragraphs...
   - Working Storage conventions
   - Procedure Division conventions, processing, rules
   - Screen headers titles, date, time, PF keys, screen/program names, error messages, help screens...
   - Report formats

I. Unit Testing

Unit testing should include the following activities basic program functions, combined data condition testing within a program, module to module linkages within a program and operational command language (JCL) for each program.

1. Establish Unit Test Environment
   This task establishes a well controlled environment for the conducting of unit test activities. This task includes allocating space for test versions of programs and their respective test data.

2. Develop Unit Test Plans
   A unit test is developed for each program. prior to execution the plan is reviewed and approved by the team leader. The test plans describe all test conditions to be tested, required test data and expected results for each test.

3. Generate Unit Test Data
   All test data sets (ie. multiple transactions with variations of data) required to complete the unit tests are created in this task. Hardcopy printouts of unit test transactions are attached to each unit test plan.

4. Prepare Unit Test Job Streams
   Required command language instructions are prepared for each program. Special utility steps to facilitate review and verification of unit test results are included as needed.

5. Execute Unit Tests
   All unit tests are executed until each has been successfully completed. Actual results are reconciled to the expected results and all program logic errors are corrected. Final test results are attached to the unit test plan for each test condition/step. Successfully executed unit test plans are reviewed by the team leader prior to movement of the program into the system test library.

J. System Testing

1. Establish an overall system test stradegy
2. Complete development of detailed test plan
3. Complete development of system test data
4. Complete system test
   - verify system test results
   - log all errors into "TEST PROBLEM REPORT SYSTEM"

K. Procedures and User Documentation

1. Indicate data preparation procedures
2. Procedural flow overviews
3. Control balancing and reconciliation precedures
4. Error correction procedures
5. Description of system interfaces
6. Key terms and descriptions
7. System closing schedules
8. Organizational structure and job descriptions
9. Finalize procedures and user documentation

L. Training

1. Review total training program
2. Develop course material
3. Develop instructor training
4. Conduct training course

M. Implementation and System Acceptance

1. Statement of acceptance test strategy
2. Statement of user and DP acceptance criteria
3. Implementation plan
4. Production job streams
5. Operations documentation
6. Verified conversion specifications
7. Converted data
8. Obtain user approval of system

N. Post Implementation Audit

1. Complete review of existing data, software and documentation
2. Conduct a detail review meeting reporting all findings

O. System Documentation

1. Model data flow diagram
2. System structure chart
3. Security concepts and implementation
4. Backup concepts and implementation
5. Database documentation
6. Program documentation
7. User instructions

In summary, if you have a big project use all of the previosly outlined procedures. If your project is small use only parts of these process as necessary. For example, start with the basics of the five stages of the Systems Life Cycle. Your approval or control points may take place at the end of any major task. Cost can be controlled by task along with manpower hour estimates and project schedules showing elapse time estimates per task.

Your organization should have a formal Systems Development Methodology that is enforced enough to be effective. You will need Program Standards that include commented schema's, data dictionary, copylib usage, screen/report format rules, and well documented programs. A tracking/reporting system to identify problem status, a responsible individual working on the task, priority that will increase the visibility of problems to all project participants. This will eliminate much of black box planning mystery of how the project is going and when its' scheduled completion can happen. Many programmers do not have the appropriate experience for the planning, executing and reporting test results. You may wish to consider someone other than the programmer to QA this part of development process. And lastly, all systems should be built with maintenance in mind, because "change" is a key element of every good organization.

TITLE: Cost Justifying a New Technology

AUTHOR: Richard Phillips
OPT (Office Products Technology)
10681 Foothill Blvd.
Rancho Cucamonga, CA 91730

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 5034

DISASTER PLANNING -- LESSONS FROM THE 1989 CALIFORNIA EARTHQUAKE

Bud Beamguard
Syntex Corporation
3401 Hillview Avenue
Palo Alto, CA 94304
415-852-1416

Computer Museum

1. A Quick Look at California Earthquakes.

California is famous for a lot of things -- Hollywood and Hewlett-Packard being two examples which immediately come to mind. It is also well known as the most seismically active area of the United States. The celebrated 1906 San Francisco earthquake and resulting fire is remembered as one of the great natural disasters in the history of this country.

Those of us who live in the San Francisco Bay Area are constantly reminded that we live in an area laced with faults, including the great San Andreas fault which runs parallel to San Francisco Bay and to which the Bay in fact owes its existence. There are other faults too, large and small: the Hayward, Calaveras, Concord, Rogers Creek, and Pescadero faults (and many others) extend like a net throughout the entire region, and are branches of the San Andreas. Hardly a year goes by without a rude warning from one or the other of these faults in the form of a good shake, resulting in cracked walls, broken dishes, frazzled nerves. And everyone is only too aware that eventually we will be struck by a "big one" like the 1906 quake. Even the smaller faults are capable of producing serious, devastating earthquakes.

Our company, Syntex, is located in Palo Alto, near Stanford University and Hewlett-Packard's headquarters. We are also only about 3 miles as the crow flies from the San Andreas fault. Our company's buildings are situated among the rolling hills pushed up by the fault on its North American side. The serenity of the site belies its long, restless geological past. We are constantly aware that the earth may at any moment decide to become active. In the years that I have worked at Syntex, we have been rattled repeatedly by small shocks, as if to remind us of the inevitable. Our entire disaster recovery efforts tend to be oriented toward earthquakes, with a glance now and then at the more usual fires and water damage; we are only too aware of the destructive potential of earthquakes.

Earthquakes pose special problems for contingency planners. They occur without warning, and efforts to predict earthquakes remain inexact at best. It is doubtful that we will ever be able to predict them with any degree of comfort. Even with fairly accurate prediction a major earthquake would devastate a given region despite all the best preparations. It is possible to see a hurricane coming, but not an earthquake.

Earthquakes also strike a wide area, and are extremely disruptive to the infrastructure of an urban area. Electrical grids, telephone networks, gas lines, communications links, freeways, and airports all are struck indiscriminately. Even food and water may not be available for some time following a quake. Beyond that, there is the psychological impact: people are invariably in shock following a strong earthquake, unable to function rationally and productively. They think first of their families, homes, and friends; the company (perhaps rightly) comes out a distant second. All of these factors make life difficult for someone attempting to put together a sensible disaster recovery plan.

2. The Events of October 17 and Following.

On Oct. 17, 1989, a Tuesday, we were hit by the strongest earthquake to strike California since 1906. There was no warning, no pre-shocks, no alert from the seismologists. The quake struck at 5:04 PM, when many people were leaving work and on the freeway system. It measured 7.1 on the Richter scale, and was centered about 8 miles northeast of Santa Cruz in the Santa Cruz mountains at a depth of 11.6 miles beneath the surface of the earth. As earthquakes go it was only a moderately strong event, and survivors of the 1906 tremblor quickly dismissed it as "no big deal". The quake was generated by the San Andreas fault and was eventually dubbed the Loma Prieta quake. It lasted about 15 seconds, during which time the energy equivalent of 100 20-kiloton atomic bombs was released. The fault rift moved about 20 inches. In San Francisco, the peak of the Transamerica Pyramid swayed about 2-3 feet each way, as did to support towers on the Golden Gate Bridge. There were 73 confirmed deaths, although it appeared at first that hundreds had died.

Any disaster generates personal stories and anecdotes. At the time the earthquake struck I was on 19th Avenue in San

Francisco, near Stonestown Shopping Center, heading north on my way home to Marin County. The car lurched forward, and I thought that I had been rear-ended (it has happened before on 19th). The car radio was on; the announcer began screaming that there was an earthquake; he unfortunately was in a high-rise in downtown San Francisco. Then I saw trees shaking, utility wires dancing up and down, and a large sign above the shopping center swaying. Most people, including myself, did not realize how strong the quake was; generally you tend to under-estimate them. The rolling lasted about 15 seconds; my car bounced around on its shocks. Radio reports soon indicated that there was major damage including deaths. The electricity failed in San Francisco, and with it the traffic signal lights. By the time we made the Golden Gate Bridge (45 minutes later) it was obvious that we were in the middle of a major disaster: the fires in the Marina District were clearly visible from the bridge, and it was official that the Bay Bridge and the Cypress Structure in Oakland had been knocked out. I found a phone booth in Sausalito and tried to call my computer room in Palo Alto; this required several tries to complete; and there was no answer. Upon arrival home (my house was still standing, but bookcases had been toppled) there were no messages on my answering machine. What was going on in Palo Alto? It was clearly impossible to drive back to find out. The phone lines were jammed as millions of people tried to call everyone they knew, despite pleas from the authorities to stick to essential calls. I finally managed to get through to Syntex Corporate Security, which assured me that at least nothing had been flattened and that there were no fires. At about 8:30 PM I got a call from one of the applications programmers who is a notorious night-owl. The HP3000 was fine. The power had been turned off at the CPU, as well as the disc and tape drives; later I was to learn that the operator (who was getting ready at 5:04 to do a partial backup) had powered everything down, as instructed by our SOPs. The programmer reported that the building was a mess, but that the computer room seemed to be in good shape.

The next morning we arrived at Syntex to find what was indeed a mess. Security was not letting anyone into the buildings until inspections had been made. It was known that in addition to structural damage there were extensive problems with asbestos contamination and with chemical spills (some radioactive) in the lab areas. At length we were able to gain access to our computer room. The power had been reliably restored, and we were easily able to recover the computer. By 10:00 AM MPE was alive and happy, but without much work to do. It can be said that not all of the computers on site were so lucky. We had to string some emergency 4-wire twisted-pair lines into one of the lab areas, but otherwise there were no repairs necessary. My Hewlett-Packard CE called to inquire about the status of the machine. By the next day, Wednesday, the computer was completely back in business. But there was one problem: the labs that we service were a chaotic mess (we had expected this to happen in an earthquake). There was little data to run, and for nearly a week the main activity on the system was our applications development team doing their usual programming.

3. Some of the Things We Did Right.

We did have a disaster plan, partly because of the paranoia about earthquakes, partly because of government (FDA) requirements. It served us well in a lot of ways:

a. A disaster recovery plan (like a will) forces you to think about the unthinkable. What is likely to happen after a disaster? What are you going to do? What can you do to get ready? How critical is the system anyway? What are your priorities during a recovery operation? We were way ahead simply because we had thought over a few of these issues.

b. We had an elaborate list of phone numbers, not only of the "disaster recovery team" but of everyone involved in running the HP3000; in fact, the numbers of anyone whom we might need to call, like Security and Hewlett-Packard. Everyone had copies of this list and hence knew what numbers to call when the moment came. I am amazed to hear stories about the lack of preparation in this simple matter, both in my own company and elsewhere.

c. The operator knew just what to do; I had seen to it that he not only was provided with some kind of SOP, but also that he had enough training to make a judgement call should this be necessary. He knew to pull the plug on the system during an earthquake, and that the system manager would (let us hope) be able to put things back together later.

d. We had done backups religiously. The system coldload tape was available and ready, and everyone knew where it was. During the recovery we had no questions about which tapes to use, where they were, or which were current. We had practiced the procedure, and knew what to do. We were ready to go in with SADUTIL if it had come to that (fortunately it wasn't needed). In fact, the recovery seemed routine. We would have lost only a single day's work had the hardware platform been lost.

e. We had gone over the computer room long ago looking to see what was going to come loose during a quake. We had secured (i.e. bolted to the wall) a number of objects like tape racks and wiring cabinets. During the quake nothing fell over, and no equipment was damaged other than several terminals in the labs which went sailing to the floor. I would add that the earthquake moved our S/70 CPU complete with two I/O bays about one foot.

f. We understood what was going to happen to our users. We had anticipated the mess in their labs, and were in a position to move our lab-based peripherals if that became necessary. We had spools of cable ready to do quick fix-ups.

g. Rather than concentrating on a vague class of "disaster" of unspecified nature, we had expected an earthquake and had prepared specifically for that type of disaster.

4. And Some of the Surprises (Good and Bad)

a. Asbestos. We had not anticipated this problem at all. It was many weeks before we were able to have normal access to our computer room. For awhile we were considering moving the computer into a temporary location, but decided against this due to the formidable problem of relocating our cabling and datacom. For almost two weeks after the quake we had to use breathing apparatus to get through the halls to the computer room. Eventually the repair crew constructed a tunnel out of plywood and plastic film; months went by before this was removed.

b. Corporate Security was caught off-guard. They were in panic mode immediately after the earthquake. During the morning following the quake it seemed like no one was in charge, and there probably wasn't. Eventually a hot-line was established, and earthquake bulletins distributed. But there was a critical lack of reliable information right after the disaster. Fortunately the Security people knew who I was, and what I did.

c. Pacific Bell (the local phone company) was a zoo. A few interesting statistics:
Number of calls attempted via AT&T into the Bay Area during the 24-hour period after the quake: 28 million.
Normal for an analogous period: 3.5 million.
Attempted calls that actually got through: 9.5 million.
Local phone calls placed during the same period in the nine Bay Area counties: 80 million.
It was very difficult to complete calls. In some areas the phone system was out completely. Microwave towers between the Bay Area and Southern California were knocked out of alignment. Like many people, we quickly found out how dependent we were on the phone system, how fragile the system is, and how much we take it for granted. We would point out too that the alternate carrier services did not come out unscratched either, nor the cellular services.

d. The City of Palo Alto owns and operates its own electrical company. In the past they have proven themselves reliable (as far as we are concerned), and after the earthquake there was little interruption of power. At least something worked right.

e. Our buildings at Syntex were constructed to be earthquake resistant. Rather to our surprise they came out rather well. Nothing had to be torn down, despite lots of broken windows, smashed brick facing, cracked and split drywall. Nobody got hurt seriously, although we were lucky: there were lots of falling ceiling tiles and (worse) falling electrical lighting fixtures. Among the worst problems generally in the company as a whole were fallen ceiling tiles, toppled bookcases, and tipped filing cabinets.

f. When the earthquake hit, everyone jumped into their cars and went home. They were extremely anxious about their homes and their families. People with kids were frantic. The last thing most people cared about was a computer and its problems.

g. One nice surprise: the HP3000 itself turned out to be the least of our problems. The actual recovery from the quake was no more complicated that a recovery from a MPE system failure. If the HP3000 had been physically destroyed, it would have been because the building had been "lost". The machine functioned perfectly for many weeks afterward, surrounded by clean-up work with its attendant dust and dirt. We were very glad that the HP3000 does not require water cooling. Hewlett-Packard throughout the emergency showed that they were concerned about our welfare.

h. It was a lot worse that I expected, considering the size of the tremor. It is disturbing to think what will happen during a really strong quake like the one in 1906: that quake virtually destroyed nearby Stanford University.

i. It was not as bad as I expected, at least from a computer recovery standpoint. We were generally pretty well prepared, and our computer room and equipment were fortunately not among upper management's headaches.

5. Some Things We Wish We Had Done in Advance

a. We wish we had assembled a set of cabling tools, including solder and a soldering iron. We needed these and time was wasted rounding them up. In fact, we would have been better off with a complete set of computer repair tools reserved only for emergencies. In a disaster situation you can't go running down to Fry's in Sunnyvale looking for some part or some tool you should have had to begin with.

b. We needed an emergency light in the computer room. They tell me it got very dark when the power went out.

c. It would have been a very good idea to have had a telephone number, equipped with voicemail, where current status information could have been left in the form of an out-going message. Presumably the system manager's phone could have been designated for this, but it was not.

6. Some Advice for People Developing a Disaster Recovery Plan, Based on Our Experiences.

   a. Most important of all, keep it realistic. Try very hard to visualize what it is going to be like immediately after an earthquake, or a fire, or a flood. Try to find some background reading on the types of disasters you may experience, including historical accounts; they make fascinating reading in their own right, besides putting you in the right frame of mind (paranoid).

   b. Plan for specific types of disasters, rather than for "disasters" in general. It is a waste of time for us in California to plan for tornados, which are almost unknown in the Bay Area. Similarly it is a waste of time for anyone to plan for "armageddon" situations like nuclear war, meteor strikes, flying saucer invasions, etc., partly because it is impossible to project the outcome of events like these. We feel that we came out well because we had planned specifically for earthquakes.

   c. Try to get specific ideas about how you can prepare for a disaster. In California for example there are a number of good books available, mostly aimed at homeowners, which give advice on preparing for earthquakes. They are filled with very specific suggestions which can easily be applied to commercial situations including computer rooms. We have found some computer-oriented disaster recovery books to be lacking in nuts-and-bolts advice; in our opinion it is this type of detail which makes or breaks you in a disaster. There was no "big picture" during the earthquake, only a mass of confusing and disjointed events that had to be dealt with one by one.

   d. Spending money may or may NOT save you. There were situations where expensive hotsites in Southern

California were of limited use due to the chaotic state of the communications grid. Maybe it would have been different had these people been dealing with a fire-related disaster. Many people similarly had off-site tape storage facilities. Had the quake been worse, or had more freeway overpasses come down, they probably would not have been able to get at their tapes. On the other hand, there are system managers who wished that they had spent a few dollars for brackets to keep their tape racks from falling over.

e. It is probably not realistic to develop an elaborate post-disaster scenario complete with meeting plans and other choreography. The would-be building for those would-be meetings may no longer exist after the disaster. Concentrate instead on providing for communications in general. The lack of centralized communication and information was our biggest problem immediately after the earthquake. Everyone was clamoring for news, and there were wild rumors. The Bay Area has been compared to London or Berlin during WW2, by people who had been through the air raids and seen the effect on people.

f. Do not assume that anyone will be around to help with the recovery. Those key people may in fact be dead or injured. Nobody likes to consider this possibility. What if you yourself were out of the picture due to injury (or worse)? Your operators may vanish into thin air. If it is critical that a disaster recovery team report for duty, you should consider what might happen if they can't or won't. It was amazing how priorities shifted after the quake.

g. Realize too that everyone (including yourself) may be in a state of shock, unable to function at their best. It was estimated that up to half of the Bay Area population suffered from moderate to severe depression following the quake, and this went on for weeks. The situation was so bad that some companies set up counseling services to overcome this problem. Similarly, many people eventually wilted due to stress and fatigue.

h. Practice your recovery procedures, and have drills to make sure that your plans actually do what they say they will; otherwise they are just vaporware.

i. If your company has become so dependent on its computer that down-time is no longer acceptable, make sure that management understands what the risk is with disasters. Don't be shy about telling them horror stories about problems in other companies. Management seems to be getting better at taking disaster recovery seriously. 1990 has been a great year in California for getting budget for this sort of thing.

j. Many organizations have gone to great lengths to protect their computers, but have forgotten the support systems upon which these machines depend: datacom networks, electrical supplies, air conditioning, building structures, etc.; the earthquake had its biggest impact on just this kind of infrastructure. It can cost lots of money to back these support systems up.

k. Luck and Murphy's Law are going to play a bigger role during a disaster than you would ever believe. Something will fall through the cracks despite your best efforts.

7. Summary

We were lucky during the quake. We may not be so lucky during the next quake, which could happen at any time without warning. I don't want to think about what would have happened to us if the Richter rating had been 8.5 instead of 7.1. I have lived all my life in earthquake country and I have been through lots of quakes minor and not-so-minor, ever since I can remember; this one will certainly stick in my mind for a long time.

If there is one most important thing I learned during the earthquake it is this: that we are vulnerable. Disasters can come at any time, perhaps by definition; hopefully we will be prepared for them when they occur.

EDI - CHANGE OR DIE! - A SURVIVAL MANUAL

James C. Himmelreich
Graco Children's Products, Inc.
P.O. Box 100
Elverson, PA 19520

Telephone: (215) 286-5951

EDI, the electronic exchange of business information between customers and suppliers, has begun to fundamentally change the way business is being done, not only in the United States but also around the world. In the midst of this cataclysmic process, the attention of both the company and its customer and supplier base focus on the EDI coordinator. The typical EDI coordinator is normally from the MIS department and has usually been exposed to EDI for only a brief time before undertaking this assignment. It is for this otherwise unsung hero of the corporate enterprise that this survival manual is written, with the hope that our best implementations of Electronic Data Interchange lie before us (not behind us).

WHY YOU NEED A SURVIVAL MANUAL

Because EDI is now a requirement of doing business

One of the most important observations that a new EDI coordinator can make is that the top brass certainly seem to take this new technology seriously, as well they should. Many large companies have determined that the effective use of EDI can give them an advantage in the marketplace, by eliminating the computer to paper to computer path of doing business (Fig. 1) with a direct computer to computer approach (Fig. 2). Notice that the number of steps involved does not change, but electronic communications are substituted for manual communications. More about this later. For now, translate "effective use" as "maximum participation". Therefore it is in the customer's interest to get as many of his suppliers doing EDI with him as soon as possible. While the supplier may not be as well equipped to do EDI as his customer, he is certainly aware that it has a direct bearing on his ability to do future business with that customer. Thus, there has been a growing realization among business executives that EDI is an emerging trend of doing business in the 1990's. That is why EDI coordinators in general are feeling the pressure both from its own company's management as well as from its external "trading partners", the suppliers and customers with whom the company does business.

Because you are now servicing new kinds of customers

As previously mentioned, both the company and its trading partners now have a direct interest in the effort and progress of the EDI coordinator toward adding and maintenancing new trading partner relationships. Recall that most MIS organizations have in the past served largely an internal customer base. This has led to a general lack of understanding of the external customers of the business by MIS, with the result that most EDI coordinators begin their role of interconnecting with trading partners with a limited knowledge of that trading partners' requirements and goals.

Because your customer has time-critical objectives

Remember that your trading partner perceives a competitive advantage to the widespread use of EDI with his firm. It has therefore become an urgent priority to many large firms to implement EDI with their largest trading partners in the shortest possible timeframe. This is accomplished by ultimatums such as service charges for non-EDI invoices after a set date or the transferring of purchase orders from businesses that do not have an EDI program in place to those firms which do have the ability to receive those purchase orders electronically.

Because your current systems were not designed for EDI

Core business systems (such as sales order processing and billing) tend to be resistant to change for several reasons. Because these are vital functions to the company, substantive changes to these systems are disruptive (at least temporarily) to the company's operations and must routinely be reviewed by the key people affected by the changes. This has the effect of making most core business system changes lengthy, expensive and tedious. The result is that many businesses tend to make minor modifications to problematic software rather than major changes. So we see that the logic processes governing these systems would be most likely to have been developed at a time when EDI was not in view, making it a daunting task to enable the system to receive electronically transferred information. For example, a sales order system may have been developed to interactively check for invalid data at entry time to prevent illogical or inconsistent data from entering the system. With the introduction of a group of electronically transferred documents to the system, a batch editing system would have to be applied and procedures for interacting with the data which failed the batch edits would have to be developed in order to integrate electronic information into your in-house system. It must be recalled at this point that your trading partner's objectives are near-term ones. This often forces the EDI coordinator into the interim step of receiving orders electronically, printing them out on a formatted document and having sales order entry personnel manually enter them into the in-house system, making EDI for the moment seem to be a cumbersome and complicated way of doing business indeed!

Because EDI standards are young and dynamic

Up to now, very little attention has been paid to the electromechanics of just how the document is able to be written by the sender in such a way that the receiver can read it. The approaches used to do this fall into two categories: PROPRIETARY DOCUMENTS and STANDARDIZED DOCUMENTS. When EDI was first attempted, a sender and a receiver would agree on how the document would look in its communicated form, thus forming a proprietary document, one which could be understood by those specific companies. It is easy to visualize how a company communicating electronically with five trading partners may have had to agree to five different ways of sending or receiving the same document. For a number of years, this very difficulty impeded the development of EDI so that few companies saw a tangible economic benefit from doing business electronically. This began to change as more generic document formats were developed so that a wider range of companies could use the same format. Organizations such as the ANSI Accredited Standards Committee X12 (known as ANSI X12), the Uniform Code Council which produced the Uniform Communication Standard (UCS), and the Transportation Data Coordinating Committee (TDCC) prompted the use of their respective standards by periodically revising the standard to make it applicable to a wider variety of business relationships, a process which continues today. The challenge posed to the EDI coordinator lies in selecting a standard (or standards) and keeping up with the frequent revisions to the standard (in some cases, twice a year).

In addition to the document being transmitted, the mode of communication (asynchronous, bisychronous, and synchronous) the baud rate (1200, 2400, 4800 or 9600) of the transmission and the character set (ASCII or EBCDIC) must all be agreed upon by the trading partners. Of primary importance is whether the communication will be direct or use the services of a value-added network (VAN) which acts as an electronic post office where sent transmissions can reside until the trading partner accesses the VAN to receive them. These issues, taken as a whole, clearly require the formation of a company strategy for dealing with EDI. However, they also require someone with both a technical and a business acumen to manage them. Because the technical issues are so numerous and complex, it has generally been the case that the EDI coordinator comes from the MIS department. In fact, it is still customary for many MIS departments to simply be "assigned" EDI as an additional service that they are expected to render to their respective companies. So while the ultimate objective of the EDI coordinator is to take on these challenges, his primary objective must be defensive in nature. In other words, he must be able to survive the demand for EDI services placed on him by both his company and his trading partners long enough to allow his company to begin to enjoy the benefits of EDI.

THE RULES OF EDI

Before a survival strategy can be presented, it must be emphasized that the rules of the conflict must be clearly understood. These are briefly stated as follows:

1. It doesn't work at all until it all works.

2. Until you can clearly demonstrate otherwise, its YOUR problem.

3. Nature always sides with the hidden flaw.

4. Failure to plan is planning to fail.

5. 90% of this is half mental.

6. All constants are variable.

Explanation of the Rules

The first two rules are a terse way of illustrating that the EDI coordinator's responsibility begins with the intent of your company and your trading partner to communicate a business transaction and does not end until that document is being communicated to the mutual satisfaction of your company and your trading partner. Anything along the way that blocks progress is your responsibility to resolve and solve. The third rule is a corollary of Murphy's Law which serves as a reminder that you are most likely to run into a difficulty in an area where you least suspected that a difficulty would occur.

Another way of phrasing the third rule might be: When in doubt, check it out! Rule four is credited to John Wooden, the former college basketball coach at UCLA, known for both his methodical coaching techniques and the great winning tradition that resulted from those techniques. Rule five is credited to Yogi Berra, a superb professional baseball player, coach and manager. Together they emphasize the importance of mapping a plan and then following the plan. You must visualize the EDI process impacting your company before you can construct a workable arrangement for doing EDI. The final rule briefly says that EDI is an ongoing process; standards change, systems get upgraded, customer needs vary and the need to reassess the trading partner relationships you have already established is at least as important as adding new partners.

THE SURVIVAL STRATEGY

Because of the dramatic effects of implementing an EDI relationship with your trading partners (or more to the point, of <u>not</u> implementing these relationships!), the survival strategy outlined below bears much similarity to a military plan. After all, military planners and strategists have been concerned with <u>survival</u> much longer than their EDI counterparts!

1. Recognize your situation.
2. Assemble an attack plan.
3. Distinguish your allies from your sympathizers.
4. Enlist recruits.
5. Draft aides.
6. Keep in sight of your supply lines.
7. Put your reinforcements where the battle is hottest.

<u>Recognize your situation</u>

This is a whole new situation for the EDI coordinator coming from a traditional MIS environment. For the first time in a long time (maybe ever!) budget has been approved (usually because of interaction between your CEO and your trading partner's CEO), and you have high visibility. In other words, you are now a high priority to both your company and your trading partners. Do not be shy about exploiting this advantage in order to accomplish EDI implementation! Do not shrink from publishing your EDI progress as it is achieved so that necessary additional resources can be gained. Make sure your sales people know how many customers are electronically ordering from you (it can be a potent competitive advantage for _them_ and for _you_).

Assemble an Attack Plan

EDI Rule #4 assures us that without planning, disaster looms. Begin by constructing a "best case" scenario. Does your company require the services of an outside EDI consultant or can your own people map out an "ideal" trading partner relationship flow? Will you use the services of a value-added network or will you be directly connecting to your trading partner's computer? Which standards will you and your trading partner use to communicate your business documents? Will you be using third-party software to do the standards translation, or will your own people accomplish this (not recommended because of EDI rule #1, #2, and #3)? Take the results of these decisions and lay out the "plain vanilla" approach for your company. Set as a goal to get the maximum number of your trading partners on this path, or as close to it as possible. This will keep your individual trading partner variations to a minimum and quicken the pace of your implementation time schedule.

Distinguish your ALLIES from your SYMPATHIZERS

An Ally is someone who can and will _help_ with manpower, finances or timeframe. Sympathizers _care_ but they can't _help_. Minimize your involvement with Sympathizers. Make every effort to get maximum contribution from your Allies. As an example, one of my trading partners was sending an EDI purchase order with certain instructions as free-form notes. We were able to persuade them to send that information to us in an encoded segment that we were in turn, able to automatically upload to our in-house order processing system. Other trading partners of _your_ trading partner can be allies, also. Determine how _they_ overcame your obstacles. Steal ideas shamelessly. Remember you are not judged on your originality but on your ability to do EDI effectively and quickly.

Enlist Recruits

Allies can be found from many non-traditional sources. Remember EDI Rule #5 - use your brain! People in your sales and accounting departments have at least as much to gain from an electronic trading partner relationship as you do. They can be excellent sources of how things get done at your trading partner's company, and who to contact if you want to accomplish something. EDI seminars and user group meetings can also be invaluable in lining up Allies. As has already been stated, you do not need more sympathizers! Maintain good relationships among your company's departments. Take the sales manager of your next EDI account to lunch. What you learn may hasten your implementation. Besides, he may even foot the bill!

Draft Aides

If you are in a situation where you have people who report to you, get them fully involved in the mechanics of setting up and maintaining an EDI relationship. Remember that good EDI personnel are made not born! If you do not want to take calls on your vacation, honeymoon or while you have a 104° fever, devote time to training your people. If you are unfortunate enough to be a staff of one, make sure to take your next vacation in a place you cannot be reached. One week should be sufficient to convince your boss that a trained backup would be helpful.

Keep in sight of your supply lines

Despite advice you receive to the contrary, do not "pioneer". Stay as close to defined standards as possible. Maximize the likelihood that someone else that you can contact tried to accomplish exactly the same thing you are attempting, thus making them an Ally. Make sure if an outside consultant is involved in the implementation process, that evening and weekend telephone numbers are made available. Remember EDI Rule #2. Also, keep the other key members of your company (especially top management) aware and alert to where the company is on the implementation schedule. Those not involved in the actual implementation process believe that EDI is "easy". Reality is your most potent weapon for counterattack - use it liberally! Publish monthly EDI transaction volume as you continue to implement trading partners. It will be a lot easier to obtain expanded resources if demonstrated results have already come from existing resources. Also, it may be helpful to let your trading partners know that you have already accomplished successful EDI implementations with others, especially if you have hit a snag due to peculiarities of that particular trading partner's relationship.

Put your reinforcements where the battle is hottest

Always ask yourself the question: **Where am I bleeding the most?** Emphasize deadline commitments. It will be much easier to postpone an implementation date if dedicated progress has already been made. Remember that you customer has no qualms about contacting your CEO if it will hasten the implementation. Document frequently what you customers expectations are compared to your own. Make sure that every possible effort is made to satisfy the customer, particularly those that know the telephone number of your boss. Keep a list of your top 10 or 15 customers. Are they all on EDI? If not, they represent the most likely source of difficult deadlines. Anticipate the interest of your major accounts in EDI and construct contingency plans to meet their needs. Forewarned is forearmed!

CONCLUSION

EDI has the potential to launch your company into the 21st century and rapidly develop your professional reputation. It can also directly lead to your unemployment! Given this state of affairs, survival skills must be acquired, honed and ultimately mastered. This paper has been developed to assist you in the development of these skills. There is a growing body of EDI information, consultants, conferences and practitioners throughout the United States and around the world. Your ability to mobilize these resources to meet your company's EDI needs has a direct effect on your company's future as well as you own.

Here's to your success!

FIG. 1

Manual Document Exchange

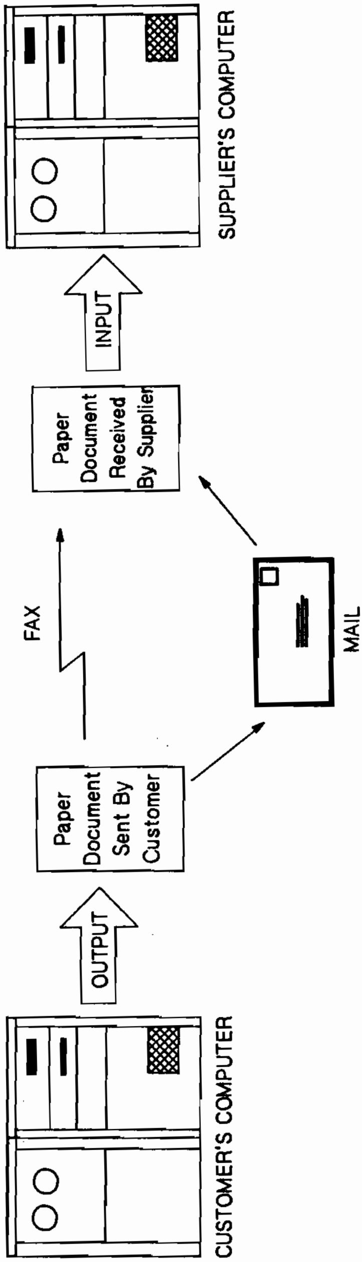
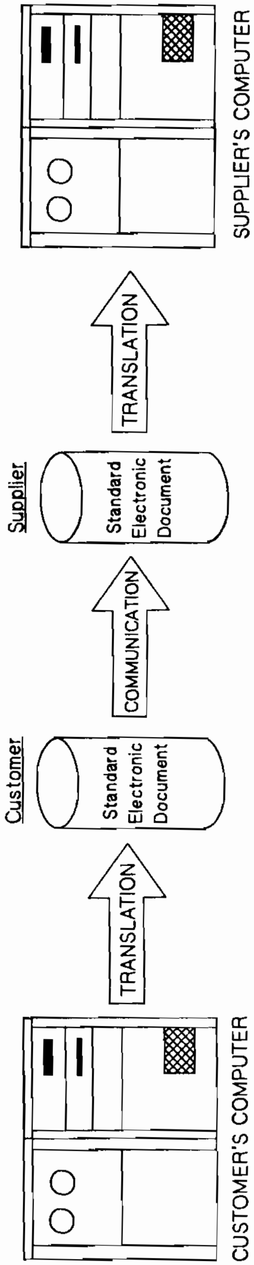


FIG. 2

Electronic Data Interchange (EDI)



TITLE: EDI.....A Programmer's Perspective

AUTHOR: Joel Kushner

Merit Systems, Inc.

901-D North Ave. NE

Grand Rapids, MI 49503

616-459-8324

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 5052

TITLE: Executive Information Systems-Realizing the Potential

AUTHOR: Mark Symonds

IISI

63 Nahatan St.

Norwood, MA 02062

617-769-7511

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 5054

How to Win at Your Next EDP Audit!

Betsy Leight
OPERATIONS CONTROL SYSTEMS
560 San Antonio Road, Suite 106
Palo Alto, CA
(415) 493-4122

As data center manager, you often walk a precarious tightrope, successfully meeting corporate objectives while directing the day to day processing activities of a busy data center. And during your daily review of the data center, some concerns may go unnoticed. These are the areas most often uncovered during an EDP audit, and they can result in the imposition of unnecessarily stringent requirements on the data center.

This paper attempts to turn a tough assignment —facing an EDP audit— into a routine exercise by focusing on those concerns most often overlooked by the data center manager. By understanding what the EDP auditor looks for during a review, you will have the knowledge to prepare your data center to pass an audit with flying colors. This paper will also arm you with a checklist for improving and automating your operations, and help you to successfully anticipate even the toughest EDP audit.

PROBLEMS ADDRESSED

During a review of the data center, the auditor is chiefly concerned with the efficiency and security of DP operations in the following areas: standards and procedures, operational work flow and controls, scheduling, data security, change control, equipment utilization and efficiency, disaster planning and recovery, and environment.

If the data center manager and staff understand what the auditor is looking for and what information is needed by the auditor, the review can proceed more smoothly, and the results can be more beneficial to the organization.

STANDARDS AND PROCEDURES

Standards and procedures that should be in place and enforced include:

* Ensuring proper timing in running programs, inserting changes into programs, and using the correct data for programs
* Protecting the data and programs from accidental or intentional destruction
* Ensuring that the data processed is complete and accurate
* Specifying methods of physically moving input and output
* Scheduling work and getting work rerun in the event of an error or disaster
* Specifying procedures for controlling data, programs, and the flow of work
* Keeping records of work performed
* Determining and recording sufficient resources for the work
* Performing maintenance and general housekeeping associated with the operation of the computer center

The EDP auditor may also want to verify that formal standards exist for systems development and maintenance, program and system change control, library operations, computer operations, and documentation.

OPERATIONAL WORK FLOW AND CONTROLS

The auditor will investigate specific items in this area, including whether:

- Input data from other departments is complete and entered on time
- The data center keeps job accounting information
- Job accounting information is evaluated and used by management

Error control procedures should also be reviewed. Specific questions asked include:

- Is anyone notified in case of a production processing error?
- Are errors documented?
- Are error statistics accumulated or ignored?
- Are errors followed up on so that they do not recur?

The auditor will also confirm that downtime is reported and statistics compiled. A log of late reports and jobs should be maintained.

There should be a formal communications channel between data center operations and other departments; operational tips and other advice should be passed to all operators.

All problems encountered at the computer, as well as any action taken to prevent their recurrence, must be documented. Operators must also receive feedback on reported problems. The auditor will verify that headers and $STDLIST information is used and checked.

Next, the auditor scrutinizes output report distribution and disposal and determines whether:

- All reports have been distributed to the proper user
- Procedures have been established to control the distribution of sensitive output
- Procedures exist for disposing of confidential reports when they are no longer required

Finally, the auditor will want to ensure that jobstream run instructions are kept up to date.

SCHEDULING

Efficient and effective scheduling is extremely important in providing a high level of reliability and predictability to DP operations.

The auditor will determine whether:

- Daily processing activities are scheduled and a daily contingency schedule is maintained. A strict schedule for nightly batch runs should also be established and adhered to.
- Actual run times are recorded for batch programs.
- This data is used to calculate expected run times to ensure that runs have not been terminated abnormally.
- Unscheduled runs are supported by a work request or other written authorization. Schedule deviations should be documented and followed up on by a supervisor.
- User-submitted jobs are recorded to allow forecasting of future schedules, resource requirements and special processing considerations for online systems.
- All jobs are submitted through or controlled by data center operations. All output should be routed by operations to the appropriate destination or picked up by the user.
- Standards cover the type, quality and quantity of forms kept on hand.

DATA SECURITY

Data base information should be protected from unauthorized access or loss. Employees must be instructed about their responsibilities concerning confidential information. Management must periodically review and update controls and security provision relating to data. Live production programs should be physically separated from development programs. The staff should be prohibited from running test programs against live files, and operations personnel should be denied access to sensitive data files.

To maintain security, operators should be prohibited from renaming or transferring programs without supervisory approval. Internal labels must be used for all data and program files.

Passwords and lockwords should be used to protect accounts, users, and data files. Passwords, lockwords, dates, and constants should be introduced at run time, eliminating the need to hard-code sensitive data into jobstream s.

Access violations must be logged and reported to the security manager. An automatic log-off feature prevents unattended terminals from posing security threats. The auditor should examine the area above the suspended ceiling in the computer room to confirm that it is accessible only from that room.

The auditor investigates blank check stock and other negotiables to determine whether they are issued on a run-schedule basis, kept in a secure area when unattended, controlled by access forms, and periodically inventoried.

Every site's security needs differ according to hardware, business focus, personnel, system function, work schedules, work environment and numerous other variables. Having stated this, there remain several constants which can help you determine your security needs. Take a moment to answer the following questions:

* Are data processing employees instructed as to their responsibilities concerning confidential information?
* Are live production programs physically separated from development programs?
* Are program library changes approved and accounted for?
* Are operators prohibited from renaming or transcribing programs without prior supervisor approval?
* Are internal labels used for all data and program files. Is an operations log maintained?
* Is the area above the suspended ceiling in the computer room accessible only from that area?
* Are blank checks and other negotiables issued on a run schedule basis only?
* Do you ever have the same password for more than thirty days?
* Is sensitive data endangered by sessions that remain on unattended terminals?
* Has your auditing firm asked for stricter reporting standards?
* Is management spending too much time implementing password changes?
* When a person who has access to sensitive information leaves your organization do you globally change passwords?
* Can users log on to any terminal?
* Are your ports being tied up by people who fail to log off?
* Can users log on to terminal from remote ports?
* Can users log on from remote ports at any time of day?
* Are additional passwords needed to log on remotely?
* Can users circumvent existing procedures to run jobs during off hours?
* Would your security be enhanced if passwords were not embedded into jobstreams?
* Do users have access to sensitive jobstreams?

CHANGE CONTROL

Change control procedures for computer programs should be established and followed. The intent of these controls is to prevent unauthorized, inaccurate, and unreliable program changes from being incorporated into the live production environment. Both scheduled and emergency changes must be appropriately controlled to maintain the ongoing integrity of software.

The auditor will check to see that the following techniques are in place to ensure that proper controls are being maintained over your program changes:

- Develop and adhere to formally approved written standards for all program changes
- Define and enforce procedures detailing who can initiate and who can authorize program change requests
- Describe and track the nature and reasons for proposed changes
- Enforce testing and acceptance procedures for all program changes including emergency changes
- Test all program changes under normal operating conditions
- Involve users in preparing test data and reviewing test results
- Investigate and correct all errors before transferring code to production
- Certify that all test results demonstrate adequate protection from fraud, waste, and misuse of the program
- Document all program changes and update appropriate documentation as changes are made
- Log all completed changes as well as those changes in progress
- Utilize a formal system to report all changes to users and project managers
- Enforce a checkout–checkin procedure that prevents a file from being simultaneously modified by more than one programmer
- Develop procedures to analyze whether other systems are affected by new program modification
- Retain and secure original source code until changes have been processed, tested and updated
- Limit the frequency of program changes, except for emergency cases
- Notify both the user and EDP project manager when emergency changes are made

EQUIPMENT UTILIZATION AND EFFICIENCY

Once it has been determined that the entire DP department is following a properly implemented set of standards and procedures, the auditor may wish to review equipment utilization.

The auditor will determine how much machine time is spent on reruns, whether reruns are analyzed, and whether certain jobs are especially susceptible to reruns. The auditor will also review programs or jobs for insufficient file design or utilization. Another area to check is the full multiprogramming capability of the system for batch production. The auditor will determine whether multiple jobstreams run concurrently and whether CPU–bound and I/O–bound jobs are mixed to maximize overall throughput. The auditor then reviews whether many jobs can be restarted without rerunning the entire job.

PERSONNEL USE AND EFFICIENCY

Review of personnel practices can be a sensitive issue. Key areas of interest include:

* Do operations personnel require extensive training and experience to be effective in processing daily production work? Is extensive knowledge of each application run necessary?
* Is there a system to schedule and monitor regular daily processing? Is the system effective? Does it operate without excessive human involvement? Or do operators spend a large part of their time tracking jobs in execution, replying to program messages, and changing job priorities? Must operators modify jobstreams at run times?
* Are all necessary tapes, forms, and other resources available when needed?
* Is there excessive turnover? Does daily production depend on specific individuals?
* Is the operations department treated as less important than the rest of data processing?

DISASTER PLANNING AND RECOVERY

This catchall category includes everything from proper insurance planning to physical security procedures. The auditor will want to determine whether the emergency plan is adequate in relation to the risk. This plan should be kept current and distributed only on a need–to–know basis.

The plan for off–site storage of files and documentation should specify:

* The conditions for use of off–site processing
* Processing priority for applications
* Resource requirements
* Job scheduling
* Run documentation
* Required tapes, forms, and supplies

Formal procedures for hardware backup should also be instituted.

ENVIRONMENT

The auditor may wish to review the work space to ensure that it is adequate for the number of employees. The environment should be neat, and supplies should be easy to locate.

Auxiliary items located outside the computer room, such as bursters and de–collators, should be accessible for the flow of work in the department. Tapes, disks, and other storage media should be stored in a closed, fire–protected, limited–access area.

RECOMMENDED COURSE OF ACTION

By understanding what the EDP auditor looks for during a review, you can now prepare your data center to pass its next audit. The checklists provided in this paper will help you to successfully anticipate even the toughest EDP audit. The data center manager should be aware that the following more general advice can also greatly enhance the data center review:

- Provide the auditor with as much information as possible
- Implement a software system that leaves clearly defined audit trails
- Keep accurate records
- Maintain formal written standards and procedures
- Implement an effective data security system and maintain an emergency plan
- Follow the auditor's recommendations and procedures in preparing for future audits to ensure efficient and cost–effective operations

IMPLEMENTING SALES FORCE AUTOMATION SYSTEMS

BY MARK P. SHIRMAN - INNOVATIVE INFORMATION SYSTEMS
63 Nahatan Street
Norwood, MA 02062

I. INTRODUCTION

Automated systems have become commonplace in almost every industry providing solutions in a variety of different application areas. There are scores of potential software alternatives for Accounts Receivable, General Ledger, Order Processing, and Inventory Control, but the whole area of automating the Sales Force is a concept that is just coming into its own. For many years salespeople have had to rely upon detailed recordkeeping and dogged determination to insure that sales prospects did not fall between the cracks. Additionally, management had very little way of accurately measuring the performance of marketing campaigns, telemarketing efforts, and their sales force as a whole. This paper will provide an introduction into the world of the fully automated sales force. We will discuss some basic philosophies as well as explore some of the features and functions that should be present in a sales lead management solution.

II. WHY AUTOMATE ?

There are literally dozens of reasons why a company should consider automation its sales force. Some of the major benefits are as follows:

- Provide a tool for the sales reps so that leads are appropriately and completely followed up

- Provide a tool for management to gain better control over not only the sales side of the business, but the entire business operation

- Provides a beneficial Return-on-Investment

- System can provide an interface to the other automated applications in place

- Allows more comprehensive and detailed marketing analysis to be performed

- Provides a method to standardize the approach to sales, so that personnel turnover can be more easily managed

- Provide an easy way to import or handle mailing lists from outside sources as well as export information to other systems or companies.

The first of these benefits, a tool for the sales reps, can easily be put into perspective through the following graphics:

WHY DO CUSTOMERS STOP BUYING?

1% Die
3% Move Away
5% Buy from a Friend
9% Buy from Competition
14% Product Dissatisfaction
68% Because of no contact or Indifference or Attitude of Sales Rep
(Source: Edward Leader, Professional Sales Seminar 1981)

These are some pretty telling statistics. In short, the more leads that a rep can follow up or touch on the more business they will generate. However, it can be an extremely difficult task to reach all the potential leads in an orderly manner, especially if the sales rep is already overburdened. An automated system can provide a rep with the tools to develop an organized approach to lead tracking.

Most good sales lead tracking software can generate tickler or action work files for reps to work off of. These files can be generated based upon follow up dates, client interest, geographic location, product interest, and so on. This provides the rep with a more focused view of what the have to do for that day or week. The lists can be used to fuel either telemarketing operations or for personalized sales calls. The average cost for a sales rep to make a call in the U.S is about $300. Based upon this fact alone, it would appear wise to be able to selectively follow up and track those leads that are of the highest potential.

How does management gauge the success of their sales force? The obvious answer is through the numbers of orders placed, however what they don't know is what percentage of business fell through the cracks or was generated from various marketing campaigns. Additionally, due to the generally independent nature of sales reps, it can be difficult to standardize the approach to lead tracking. This can become a serious problem when an individual leaves their job. Quite often valuable time and energy is lost as people try to figure out what the rep has left behind, if they did leave something behind.

An automated sales lead tracking package can assist organizations with many of the problems mentioned above. Most packages will be able to track the origin of a lead. This can be extremely helpful, especially in evaluating the success of a particular marketing/advertising campaign. The typical example would be an ad that generates a lot of "bingo card" type leads, or requests for product information. An automated system can help determine what percentage of those leads actually led to a sale as well as what the overall potential of these were. Additionally, lead tracking could be a result of a major campaign.

Management should view sales force automation as a tool, n the same manner the sales rep does. By having access to all the leads for an organization, management can accurately create forecasts, measure rep performance, and standardize its approach to lead follow up. The key here is the increased level of control automating the sales force can bring to management. In this way it is no different than automating a manufacturing facility. Organizations often pursue this path in an effort to control costs and analyze more data, and it is no different with sales lead tracking systems.

Management should also be impressed with the return-on-investment figures that usually correspond to automating the lead tracking functions. The following pages represent some models that can be used to evaluate the ROI figures. We will start small and think about a company with annual sales of one million dollars and a software package that costs $14,400. You can see that even at a 5% increase in business the ROI figures are amazing. That 5% figure is much lower than many statistical surveys have shown. In fact Sales and Marketing magazine found productivity increases averaged 43%(Summer 1987) with the introduction of an automated sales lead tracking system. The ROI figures coupled with the benefits mentioned before make a pretty compelling argument in favor of automating the sales force.

III. PHILOSOPHIES AND APPROACH

There is a school of thought that believes that sales force automation is best achieved through the use of microcomputers. We do not believe this to be the case especially in the HP environment. A centralized IMAGE data base allows sharing of information and files much easier than with PC's. One of the goals of sales force automation is that of management control. By having a central receptacle for data, this is more easily achieved. The central data base concept can also have implications as to how the system can interface with other automated applications. MIS should not view the sales lead tracking data as simply and "island of information". It is possible to feed order entry as well as accounting systems, and manufacturing systems the information generated by an automated sales tool. Conversely, the sales software should be able to "import" information from other data base applications. This entire concept will allow easier generation of management reports and development of inquiry subsystems.

Eventhough we advocate a centralized data base approach to sales lead software, the package should have the ability to "export" information to other processing options, i.e.. another HP3000, Vectra, or Portable. The concept can be thought of as distributed processing with an interface to the host. Lets consider an example. A sales rep is in the field. He/she downloads a list of leads from the corporate office to a PC. The leads are managed, tracked, changed, and more are added. At the end of the day, week, month or whatever they upload those leads to the corporate office. The leads can still be maintained on the remote workstation, but the home office can evaluate them for forecasting and other analysis.

Return on Investment Example

Software Calculation Examples

(<u>Manufacturer</u> Profile Assumptions)

	<----Increase in Sales Due to----> Automation				
	5%	10%	20%	30%	40%
Case: $1,000,000 Annual Sales					
1. Increase in annual profit	15,000	30,000	60,000	90,000	110,000
2. Annual suuport fees	2,400	2,400	2,400	2,400	2,400
3. Software	14,400	14,400	14,400	14,400	14,000
(1. minus 2.) divided by 3. % = RIO					
	90%	191%	400%	608%	747%
3. divided by (1. minus 2.) times 12 = Payback in months	14	6	3	2	1

Return on Investment Example

per $000 in Annual Sales Volume
(<u>Manufacturer</u> Profile Assumptions)

	Current Base Amount	(Choose one below) <----Increase in Sales Due to----> Automation 5%	10%	20%	30%	40%
Sales	$1,000	$1,050	$1,100	$1,200	$1,300	$1,400
Manf. Cost of Goods	500	525	550	600	650	700
Gross Profit (50%)	$ 500	$ 525	$ 550	$ 600	$ 650	$ 700
Selling Expenses (20%)**	200	210	220	240	260	280
General & Admin. (Fixed)	200	200	200	200	200	200
Net Profit before Taxes	$ 100	$ 115	$ 130	$ 160	$ 190	$ 220
Net Increase over base	---	15	30	60	90	110
Percentage Increase over base profit due to Software	---	<u>15%</u>	<u>30%</u>	<u>60%</u>	<u>90%</u>	<u>110%</u>

** Worst case assuming all selling costs increase as the sales productivity increases. In actuality, some of these expenses will remain fixed at a lower level.

As with any other system, the software functionality should be the primary driver as to what system should be put together. The functionality of sales leads tracking software is different from many systems that may go into a data center. It is important that the projects aimed at decisions of what packages to implement, incorporate the sales force in their process. Because sales leads tracking is a relatively new concept, not many MIS people understand all the ins and outs. It is currently a speciality unto its own. Before implementing a new system make sure that all the bases have been covered with and individual that knows these types of systems.

The functionality that needs to be present in sales force software is going to vary from organization to organization. There are some basic pieces that should however be present in every one of those systems. Most importantly the system must be easy for the sales rep to use. This implies more than just user-friendliness. Flexibility, speed, and as few keystrokes as possible are all vital components of this. If the system is not easy to use, you will never get your sales reps to use it. The more the reps use the system the better sales results will be and the more the overall data will be helpful to management.

Some additional features/functions that should be present in an sales lead tracking package are:

- Ability to import mailing lists from third party or outside sources
- Creation of tickler/action fields to zoom in on only relevant data
- Ability to store free-form text easily along with lead information
- Ability to perform literature fulfillment and mailing labels for mass mailings
- Should be able to generate batch letters as well as customized letters on the fly
- Should have the ability to look at data a variety of different ways through keys

- Ad-hoc reporting features

- Flexible security so that there is no problem with reps stealing information from one another, but management can view all relevant information

- Ease of interface into other existing data processing applications

These are just a few of the major things to look for in a sales automation package. There are many more things which could be of importance to your own shop i.e., interface with graphics, audit trails, data dictionary, documentation...Just remember flexibility and ease of use are of primary importance.

CONCLUSION

It is our that the time to automate your sales force is now. Those organizations that do will have a competitive edge over their rivals. Most of the other major functional areas in an organization have been automated, the sales force and lead tracking areas represent one of the last frontiers in data processing. The ability to get a handle in an organized way of how your reps are performing, while at the same time giving your sales force a powerful tool, is a temptation all MIS directors should take seriously.

Improving Software Quality

by Robert Green

Robelle Consulting Ltd.
8648 Armstrong Road, R.R. #6
Langley, B.C. V3A 4P9 Canada
Telephone: (604) 888-3666
Fax: (604) 888-7731

I am founder of a small software company that makes tools for the Hewlett-Packard 3000 computers. I also spent seven years working in the HP factory and three years in an end-user MIS department. I have been involved in the development of applications, operating systems, and tools, some of high quality and some of low.

Our software products QEDIT and SUPRTOOL have high reputations for quality in the HP 3000 marketplace, but I suspect that our initial design and code are not much better than other people's. What may be different is the way we evaluate and revise our efforts before official release, and the way we react to error reports after the release. The result is that few users ever experience our programming or design mistakes. When one does, we have systems in place that can correct most of them quickly. That is the topic of this paper.

I have good news and bad news regarding software quality. The bad news is: compared to hardware, software quality is terrible. The good news is: I don't think it is the programmers' fault. The techniques for writing good code are well-known and I won't rehash them here. No doubt there are still more techniques and tools to come that will assist in this difficult task. However, most of the quality problems occur because of mismatches between the programs and the needs of clients, not because of bugs.

Outline

Quality Equals Superior Value to the Client

Involve the Client From the Start

Start Small and Make Constant Improvements

Solve the Actual Problem First

Know Thy Client

Admit That Mistakes Are Inevitable

Concluding Remarks

Quality Equals Superior Value to the Client

The market value of a product is <u>not</u> an intrinsic value, not a "value in itself", hanging in a vacuum. A free market never loses sight of the question: of value to <u>whom</u>? [Ayn Rand]

Defining Quality

A good way to start any inquiry is to define your terms. According to my *Little Oxford Dictionary*, **Quality** is a noun meaning "degree of excellence". **Excellence** is defined as "surpassing merit", **Merit** as "goodness", and **Goodness** as "virtue". So what we have here is an ethical issue: **Quality is the relative virtue of a thing, compared to alternatives.**

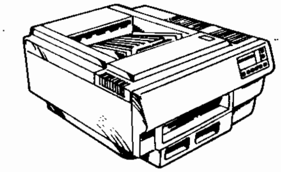
Your software has quality to the extent that it provides **Value** to some living, breathing people with choices and options. If another program solves a similar problem in a way that the person values more, it has higher quality.

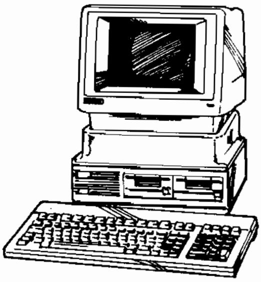
Quality Demands Constant Attention

Quality is a hot topic right now with business gadflies like Tom Peters: "**Quality improvement is a never-ending journey.** There is no such thing as a top-quality product or service. All quality is relative. Each day each product or service is getting relatively better or relatively worse, but it never stands still. Ford is doing well now, but Toyota, which Ford sees as its principal competitor, is implementing 5,000 new suggestions a day."

Quality is Different From Correctness

Quality is not the same thing as "Correctness", which is producing a program that exactly implements the design specifications. What if the design does not specify what the clients need and want? Quality is not static: people's needs and situations change over time, and as they do, the quality of your program will change as well. You cannot separate the quality of a product or service from the evaluations of the people who will be using it at a given time and place.





The 150 "worked", but was incompatible with other PC software, used an odd-size diskette, and customers did not want the touch screen. The LaserJet, on the other hand, satisfied customer needs perfectly. [Kathy McKittrick]

Quality is Judged in a Context

Don't ever forget that the quality of your programs will be judged in the context of an installed client system. The client has other programs that he uses and understands, he has administrative policies that may differ from yours, he has people with different backgrounds and training from yours. You must be aware of the client's total experience with your program.

Clients have their own context from which they look at our software:

> **Why do some subsystems require "EXIT", some "E", and some "EX" to exit? Even in MPE XL, in a single subsystem, SYSDIAG, part of it requires "EXI" and the next level up requires "EX". I know that there is an explanation and it is because parts of the software were written by different teams in vastly different geographical locations, but it seems to me that an elementary standard such as exiting the program should be uniform.** [John Dunlop, *Interrupt* magazine]

Involve the Client From the Start

After the state spent $20 million and nearly seven years trying to computerize its public-assistance program, the first caseworkers to use the system made their own discovery: They could figure out a client's benefits faster by hand than with the computer. [*Seattle Times*, May 19, 1989]

State bytes off more than it can chew

DSHS to scrap computer system

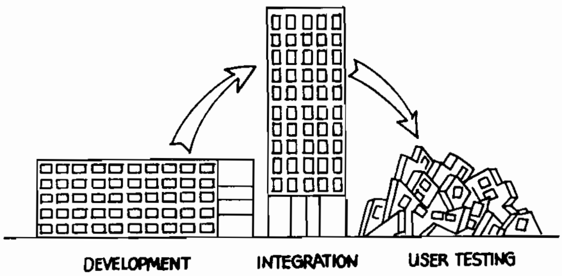
Why Do Big Projects Often Go Wrong?

COSMOS was a gigantic government project to save money calculating eligibility for social assistance such as food stamps and welfare. "After spending over $20 million, the Washington State government decided to swallow its losses and terminate COSMOS. A consultant's report recommended scuttling COSMOS. The report cited poor management, an overly complex design, difficulty to use by caseworkers, and the use of untested software." [*Seattle Times*]

This software disaster includes most of the things that can be done wrong. The state contracted with an outside firm to design and implement the system. What started as a Big Project, grew into a Giant one. State officials bragged that COSMOS would use artificial intelligence. It was seven years before the first pilot installation, when workers found it took up to twice as long to figure out a client's eligibility with COSMOS as it did manually.

The traditional method of developing a DP system includes endless user interviews, voluminous specifications, official approval by confused users, programming phase, integration phase, testing phase, user training, and endless bureaucracy. Notice that this method does not deliver any working programs to the clients until the very end. Creating a DP system without delivering **anything** to the clients until **everything** is done is like constructing a complete office building on its side, then trying to lift it into position.

"TRADITIONAL METHODS APPLIED TO AN OFFICE BUILDING"



Projects Must Be Grounded in Reality

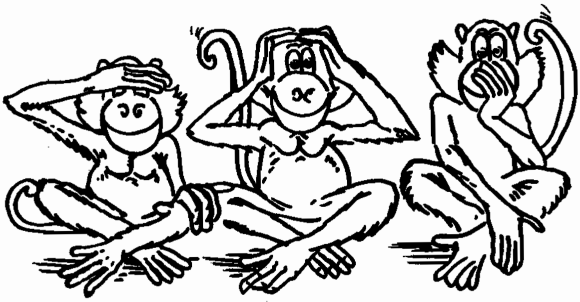
Physicist Richard Feynman was a member of the Presidential Commission that investigated the crash of the Challenger shuttle. He concluded that NASA management exaggerated the reliability of the shuttle to the point of fantasy, then regularly and subtly reduced safety criteria to maintain the published launch schedule.

> **For a successful technology, reality must take precedence over public relations, for Nature cannot be fooled.** [Feynman]

Fantasy by top management has a devastating effect on employees. If your boss commits you to producing a new accounting system in six months that will actually take at least two years, there is no honest way to do your job. Such projects usually appear to be on schedule until the last second, then are delayed, and delayed again. Management's concern often switches from the project itself to covering up the bad publicity about the delays.

> **Information from the bottom which is disagreeable is suppressed by big cheeses and middle managers ... Maybe they don't say explicitly, "Don't tell me," but they discourage communication ... it's a question of whether, when you do tell somebody about some problem, they're <u>delighted</u> to hear about it. If you try once or twice to communicate and get pushed back, pretty soon you decide, "To hell with it."** [Feynman]

An objective project goal unleashes people's minds to discover solutions and attain the goal. An irrational goal just short-circuits the best within them.



Moon's Maxim

Why is it that systems designed with great care, using experienced analysts and the latest design techniques, can totally fail to solve the client's problem? I heard one answer recently:

> **Moon's First Maxim: The process of developing a system uncovers information about the system that <u>no one</u> could have known at the offset.** [Richard Moon]

Users are not, and cannot be expected to be, systems analysts. And systems designers cannot think like users. The client often cannot describe what he wants -- he does not realize how important exceptions are. Even when the analyst extracts all his wants from him and defines them in an enormous specification, he has no idea what is critical and what is frosting. In an attempt to wrench precise specifications from the client, some shops spend so long on the design that by the time they are done, the client's needs have changed.

The clients are often shut out after the general design phase. They are asked to approve the specifications so that the programmers can get to work. One thing you can be sure of: the clients may not be able to tell you what they want, but they can tell you what they don't like when you finally deliver the code.

The Hewlett-Packard Company has more success producing quality electronic instruments than quality software products. I think the reason is that the engineers who design instruments are basically their own clients -- they can visualize what would make a better product because both they and the client are engineers. The programmers who create financial accounting packages are not accountants, they are programmers -- and they can't afford to trust their personal judgment of an accounting program's quality.

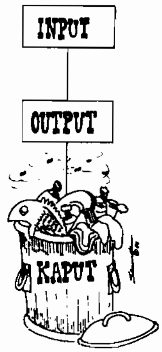
Get the Program to the Client

It was Michel Kohon who first pointed out to me the reason why it is difficult for the client to visualize the result of a program, especially an interactive one:

> **A program is not static. The actions it performs vary dynamically, depending on the information that is entered. It is a moving body and is unlikely to be adequately described without using jargon. The same applies to mathematics or astronomy, or films. How can we visualize a film from a script? This is why the sooner you show the program to the user, the better it will be for his understanding.** [Michel Kohon]

You must get the program into the client's hands in order to find out what you don't know! Once you get a reaction from the client, you can revise the program to meet his exact needs.

> **Moon's Second Maxim:**
>
> **Development methodologies that do not support iterative development are doomed to failure.**



This is a key insight. You can never get a software design correct by just studying and interviewing. You will always have to go back and revise the design as soon as you start implementing. We use a form of iterative development called the **Step by Step** method that was proposed in a 1980 paper written by the above-mentioned Michel Kohon.

Start Small and Make Constant Improvements

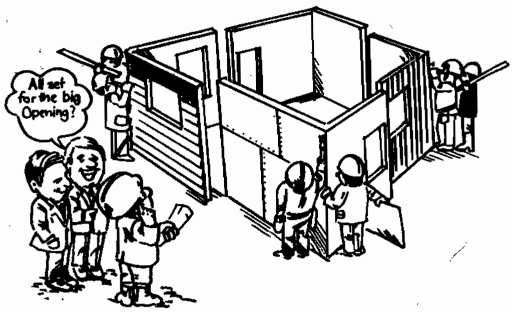
> **American companies also have often lagged behind their overseas competitors in exploiting the potential for continual improvement in the quality and reliability of products and processes. The cumulative effect of successive incremental improvements in and modifications of established products and processes can be very large; it may even outpace efforts to achieve technological breakthroughs.** [*Scientific American*]

The Advantages of Small Projects and Pilots

I have observed that small teams seem to produce quality results more often than large teams. Others have observed the same:

> **At Pacific Bell, a system was required for automating a million transactions. Two estimates were received, one from a big, outside firm (three years, $10 million) and one from a major Pacific Bell unit (two years, $5 million). Meanwhile, three South California employees took a crack at the task--and did it in sixty days for $40,000.** [Tom Peters]

Small projects have the advantage that they can be cut off or modified quickly. Big projects are hard to cancel, because of the political flak over all the money already spent, and are hard to modify, because of the rigid planning that goes into them. To be successful in the free market, you need to respond instantly to new facts and opportunities.



Small projects, especially pilot projects, are perfect for testing new ideas in the real world. Even large goals, such as a new aircraft design at Boeing, can be done as a series of small projects. Parts of new aircraft are tried out as redundant systems on current aircraft. According to Tom Peters, the key to successful innovation is to "test it now, at least some piece of it, in the real world..."

> **Is this approach 'soft'? NO! It is hard--the very hardest. It is rational and it is 'scientific'. In fact, it amounts to the organizations embracing the essence of the scientific method--empiricism and the experimental method. Piloting is the approach based on data. By contrast, decision making by proposal churning is whistling in the wind; it is the truly soft and ultimately less rational route.** [Peters]

Step by Step: Limit Time as Well as Staff

Everyone is familiar with Brook's Law:

> **Adding manpower to a late software project makes it later.**

Why is it that increasing the resources never seems to get the work done faster? One reason is economics. To produce programs, you will assign programmers, but there are never enough. Why? Because the client's demands will always increase to match your supply of programmers.

This is a common result in all human interactions. When they opened a new freeway in Vancouver recently, a highway expert said not to expect any lessening of traffic on other routes. The reason: by making it easier to travel downtown, the new freeway would entice more suburban motorists to take trips. The traffic expands to fill the roads available.

> **The only logical way to escape this dilemma is to limit the offer [i.e., supply]. How can we do that? One way is to limit resolutely the number of programmers working on a project. A second way is to limit explicitly the amount of time allocated to a program or system.**
>
> **Let's imagine for a moment that we've said we have two weeks to program our system with the existing manpower. No more than two weeks. How can we best solve the problem in the amount of time given? The natural way will be to put on paper what the MUSTS and the WANTS are. If both can be produced in two weeks, we will program both, but that is unlikely... The most important objective is to find the absolute MUSTS which can be produced with the <u>current</u> staff in a limited period of two weeks....Never go back on the two weeks allowed. It MUST be done in two weeks. Try to imagine that in two weeks' time, it will be the End of the World. Users will laugh, but they will, as well, appreciate your concern.** [Michel Kohon]

The Step by Step method suggests dividing projects into two-week chunks, then delivering each chunk to the client for actual use. This has a number of useful results. It involves the client directly and enthusiastically in the design of the system, it means you never have to write off more than two weeks' work if your design is wrong, it means you can make constant adjustments in your goals as you get realistic feedback, and it eliminates the difference between the development and maintenance programmer. Everyone becomes a maintenance programmer, charged with delivering increasing value to the client in each step.

Continual Improvement

You can learn a great deal about what produces successful software by looking at the HP 3000 software market. The HP factory often appears to follow a method that can be summarized as Design It, Code It, and Forget It. The result is that most of their software products over the last ten years have been unsuccessful. The ones that succeeded, such as COBOL, have had constant enhancements over the years.

The software products that have not had attention after first release are now forgotten. The most successful HP program, MPE, has been continually enhanced for 15 years. But consider the Spooler. The new Native-Mode Spooler is the first enhancement since the SPOOK program, over ten years ago, and the 2680 printer support, over five years ago. No wonder four vendors can make a good living selling enhancements to the Spooler.

If you want to see the tremendous power of continuing, unrelenting, tiny improvements, you just have to look at the Japanese success in manufacturing:

> **The Japanese treat every product as an ongoing experiment and are constantly engaged in improving it.** [Peters]

> **Find out what was wrong, try to understand why it had gone wrong, and then break down the corrective process into modest steps.** [David Halberstram, *The Reckoning*]

I feel strongly that a programmer should stay on a program through its life. In many shops, programmers are treated like commodities and shifted from project to project frequently. This ignores the benefits that come from continuing to work on a single project: experience in client needs, experience with the code and data structures, working relationships with the other people involved on the project and in the industry. These are lost if a programmer is shuffled off to another project as soon as he completes part of one project.

Solve the Actual Problem First

> **Identifying the long-term and short-term objectives will permit you, with the users, to draw a line of actions within an overall strategy. You will move from point A to point Z through points B, C, D, ..., with each point being an objective. But how to order these points? To provide a solution to the top problem [first] means that you will give the maximum result in a minimum of time, and you will repeat this with each successive point. Order the objectives from the maximum payoff to the minimum. These will be your Steps.** [Michel Kohon]

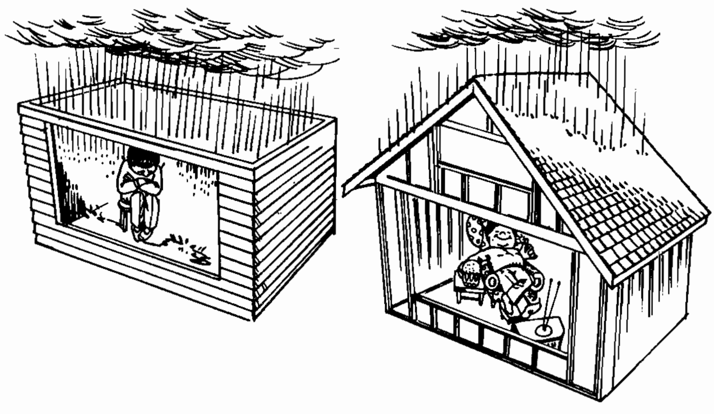
Let the Value to the Client Order the Steps

I have proposed involving the client in the design process, starting small, and improving a system in continual steps. However, what do you work on first, and how do you set priorities?

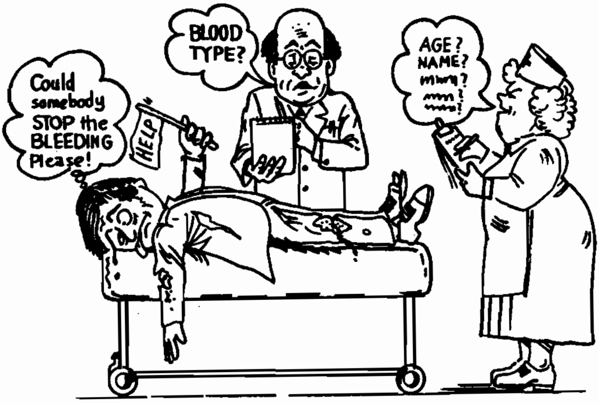
Programmers have a tendency to want to work on the technical challenges first, since that is what they know best. But a beautiful screen doesn't help the client unless it has data on it that are important to him. This is not Step by Step.

Step by Step aims to discover the client's actual requirements and program them all, eventually. Suppose a client is having cash flow problems? He asks you to provide an order processing system, expecting that the more efficient invoicing will bring in cash more quickly.

The typical response is to give him an order p'ocessing system. If you could provide a complete working order processing system in two weeks, including invoicing, you would indeed solve his cash flow problem. But you can't, so you conduct a long study and install order entry as phase one. This is more work for him and does not solve his most pressing problem. This is not Step by Step.



Step by Step challenges you to deliver something in the first step that will make a big contribution toward solving the client's most pressing problem. This is not easy to do -- it takes creative thought. You might automate just the invoices with the largest dollar amount. Or just the simplest ones, leaving the staff free to deal with the ugly invoices manually. Think of solving the 20% of the cases that generate 80% of the benefit.



Solve Part of the Actual Problem in the First Step

Until you deliver a program to the client, you have not accomplished anything, and you haven't started receiving the objective feedback that will ensure a quality system. The advantage of going after the immediate problem first is two-fold: it gets the client on your side, and it uncovers facts about the problem that may make the rest of the project simpler or unneccesary.

This is the hardest part of the Step by Step method, the part that requires the most demanding thought on your part. You will need to analyze the client's problems sufficiently to make an objective hypothesis identifying the most critical problem. For complex clients, this could be a major study. But you must always remember that the goal is the programs, not the investigation.

Know Thy Client

> **Since you're not the customer you have no way of knowing what's important and what's not important about the product.** [anonymous HP client, quoted in "HP Corporate Quality 1989"]

From the Client's Point of View

A program is inseparable from the manual, sales brochure, packaging, delivery, training, support, and installation that come with it. In the same way, an application system is inseparable from the operating system that it runs on and the 4GL that it is coded in. All of these elements go into creating the client's experience. It takes a total team effort to ensure high quality.

For example, disposable contact lenses come in a plastic package that keeps them sterile and moist until use. However, peeling off the foil seal leaves a sharp edge that can cut your hand. The contact lens may give perfect sight, the marketing may be superb, the sales team helpful, and the product distribution speedy, but if the client cuts his hand opening the package, that undercuts the quality of the entire product.

It is frustrating to dedicate two years to a computer program, as happened to a former HP employee of my acquaintance, then find that the company bureaucracy cannot deliver the program to the users for another three years. The best people quit and form their own companies -- the ones who stay often shrug and say "that's not my job." At Robelle, we hold staff meetings every two weeks in which sales, support, finance, R&D, marketing, and administration provide each other with a global picture of product concerns and customer priorities.

Break Down Barriers to Clients

Programmers Should Take Technical Support Calls. There is nothing like hearing directly from an irate user of a piece of software that you wrote to motivate you to improve it.

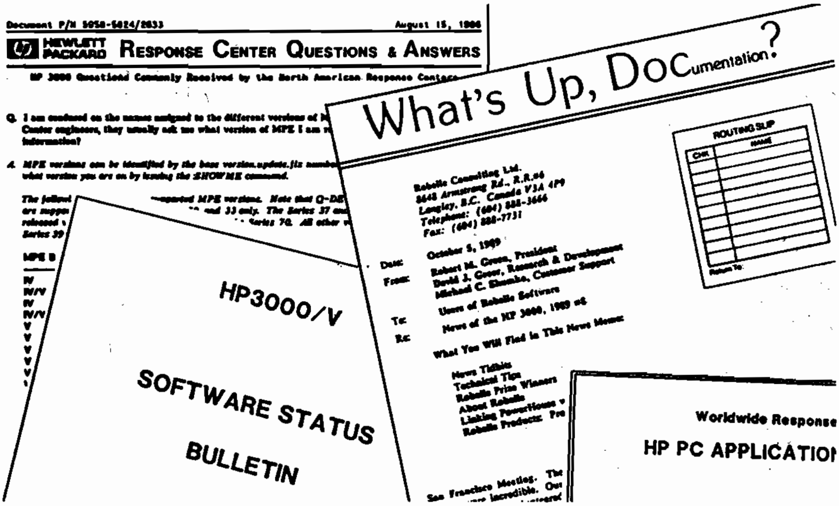
No Matter How Much It Hurts!

Programmers Should Visit Client Sites. When you are on site, people mention problems that irritate them but which they won't call about. You see them use your product in unexpected ways and use ingenious workarounds for unsuspected design flaws. Users group meetings are another good place to meet clients.

Bug Tracking Software. There is nothing worse than ignoring a cry for help. Once you have a reputation as a "black hole" (bug reports are never heard of again), clients stop looking to you for solutions, or products. We use electronic mail and a keyworded database to collect trouble calls and route them to interested parties, including programmers. Anyone can append comments to the original call report and the comments are distributed via E-Mail also. This "conferencing" makes a lab programmer feel almost as if he is on the technical support line.

```
Ref:         288
Company:     RacRillan Cloedel Inc.
Caller:      Bobin Bennie
Phone:       (904) 661-8544
Serial#:     3100
Subject:     Batch/NS/Eating job lines
Product:     QEDIT 3.7
Status:      Open

New:  DAVID GREER        Batch/NS/Eating job lines  22 Mar89 10:36

* Orig:  DAVID GREER                                  22 Mar89 10:36
Bobin has installed Qedit 3.7 on two of their development machines. Qedit
is failing in many job streams, because it is trying to do teminal
recognition when run from a remote job stream. I'm sending Bobin Qedit
3.7.1 which we belive fixes the problem in most cases. Do we know whether
we correctly detect batch-mode when remotely run from a batch job in all
cases?
```

Continuous, Intelligent, Honest Communication. A dependable, concise technical newsletter builds client loyalty. It should be selective. Clients are busy and don't have time to read everything they receive every week. The HP Software Status Bulletin contains every known problem, but doesn't highlight the disastrous ones. This would require a senior person as editor. A newsletter should be timely. If you insist on a glossy four-color format that makes production stretch out to weeks, your news will be dated. We aim for one-day turnaround.



Admit That Mistakes Are Inevitable

> **The most important lesson to be learned from this incident is that even highly talented programmers make disastrous mistakes. Morris's program was comparatively small and simple, and had a limited and well-defined purpose. It faced no serious obstacles in the form of security barriers that attempted to foil it. Morris is known to be extremely skillful, was highly motivated to write an error-free program, and was not working under the pressure of any deadline. Despite all these facts, his program contained a catastrophic error [causing it to replicate and propagate itself far more rapidly than he apparently intended].** [Ornstein]

I found this example in the ACM's comprehensive review of Robert Morris's famous Internet Virus that invaded 6,000 Unix systems on November 2, 1988. Producing quality software is difficult and mistakes are inevitable -- not just mistakes in programming, but more disastrously, mistakes even in identifying what the client needs.

Be a Humble Manager

One of the key papers in the history of structured programming was *The Humble Programmer* by Dijsktra. Most programmers have accepted his approach:

> **I now suggest that we confine ourselves to the design and implementation of intellectually manageable programs... We shall do a much better programming job, provided that we approach the task with a full appreciation of its tremendous difficulty, provided that we stick to modest and elegant programming languages, provided that we respect the intrinsic limitations of the human mind and approach the task as Very Humble Programmers.** [Dijsktra]

Now we need to extend those insights to the equally difficult task of managing "what" programmers program. Humble management is like defensive driving. You must assume that something will go wrong at the worst possible moment and be prepared to switch gears quickly.

Some Techniques for Managing Error

Explicit Design Criteria. Programmers need guidelines to help them make difficult tradeoffs while programming. At Robelle, our criteria are reliability first, then compatibility, performance, and finally features. Without such leadership from management, programmers cannot be expected to produce a consistent and dependable style of program.

The Development Diary. This is a computer file that acts as a lab notebook; it records your thoughts and plans as you work on the code. We make entries for each day, with the most recent day at the start of the file. Other sections of the file list outstanding bugs, enhancement requests, patches for known problems, and documentation problems.

Batch Jobs to Test for Stupid Mistakes. If you touch the code, you may delete a line by mistake. We do automatic batch regression testing of each new version. The tests are designed to abort if anything goes wrong, or update a results file if they make it to the end. We often run the test suite every night to check that day's changes. Ideally, each bug uncovered should be verified with a batch job that reproduces it. Once the bug is corrected, the test will pass and will ensure that old bugs do not creep back in by accident (this has happened to us!).

Frequent User Testing. The biggest danger is that you will deliver a working system that doesn't do what the client needs or wants. Even when your code does what they want, it never does it exactly right on the first pass. To minimize this risk, we send pre-releases of revised software to selected clients about once a month. A pre-release program is much like a regular release, including updated manuals and on-line help. We seek out clients who will treat this software as harshly as they would an official new product, and we often find them in our tech-support records. The benefit is mutual: we get objective feedback, and the clients often get very quick solutions to their problems.

Measure Quality in All Functions and at All Levels

What you measure is what you care about - people sense this. Measurements are good for recognizing achievements, not just detecting problems. Most jobs are repetitive -- it isn't easy to appreciate all the work you did in a year without numbers ("I filled 200 client orders", or "I wrote 2000 lines of code that got into production").

Start with concrete, mundane measurements that are easy to collect. They should be simple, they should relate to your long-range goals, and they should be understood by the people who make them. Just the fact that you are measuring will make a difference. If you show you care, people will refine the categories and numbers over time. Here are some you can start with:

Time to get a new release to the client, to fix a bug, to answer a fax.
Number of bugs reported each week.
Warranty costs, number of returned products.
Number of products shipped per day, percentage shipped in 8 hours.
Number of times competitor mentioned favorably!

Dan Warmenhoven has said that HP has two new metrics it will use to measure software quality: "... the number of post-release defects in the first year ... and the number of critical and serious open problem reports." [*Interrupt* Oct. 89].

Where Does the Money Flow in Your Organization?

Do salesmen get bonus trips to Hawaii and cars and fancy offices and awards and special dinners, while technical support people get a certificate of appreciation? Is it any wonder that the products don't work right? It isn't what you say that matters, but what you do. How is the budgeting arranged? Is there funding for on-going improvements of the software? Is your super-programmer allowed to work on a single project for years until it is a coveted tool of the clients, or is he constantly shifted to another project as soon as he delivers the first version, and replaced by a junior maintenance programmer?

Management communicates priorities every time it does anything -- whatever management spends time on and rewards is what the company will emphasize.

Concluding Remarks

The task of improving software quality is primarily a management task, not a technical one. The problem is not that we haven't adopted the latest, "perfect" system development methodology -- the problem is that we haven't been realistic about the immense difficulties in producing quality software, and we haven't been paying attention to the mundane, practical details of producing software that does what the client actually needs and wants.

I think that the primary responsibility of managers, the fundamental one around which all others will revolve, is to break projects into manageable steps, then deliver the new software produced by each step into the client's hands, so they can give you objective feedback for the next step.

References

Berger, S. et al, "Toward a New Industrial America", *Scientific American*, June 1989, Volume 260, Number 6., by Suzanne Berger, Michael Detouzous, Richard Lester, Robert Solow, and Lester Thurow.

Brooks, Frederick P., **The Mythical Man-Month**, Reading: Addison-Wesley, 1975.

Dijkstra, E., "The Humble Programmer", 1972 Turing Award Lecture at the ACM Annual Conference, Boston, on August 14, 1972.

Feynman, Richard P., "Personal Observations on the Reliability of the Shuttle," **What Do You Care What Other People Think?**, New York: W. W. Norton, 1988.

Halberstam, David, **The Reckoning**, New York: Avon, 1986.

Kohon, Michel, "Introduction to Step by Step", **SMUG II Proceedings**, Langley: Robelle, 1982.

Ornstein, Severo M., *Communications of the ACM*, June 1989, Volume 32, No. 6.

Moon, Richard, "Managing 4GL System Development in the 1990's", *Conference Proceedings of the HP Computer Users Association*, Brighton, England, July 1989.

Peters, Tom, **Thriving on Chaos**, New York: Harper and Row, 1987.

Rand, Ayn. **Capitalism: The Unknown Ideal**, New York: New American Library, 1966.

Schlender, Brenton. "How to Break the Software Logjam", *Fortune*, September 25, 1989.

Simplifying Migration with Standards: A Case Study of Effective Team Management

Dianne Willems
Hewlett-Packard
Corporate Offices
P. O. Box 10301
Palo Alto, CA 94303

This paper is a case study of a group of people working together towards a common goal. It is not a technical document on MPE/XL; nor is it a step by step set of instructions on the migration process. It is a true story about the synergy produced by a small team committed to companywide migration to MPE/XL.

The team is the MPE/XL Standards Committee and I am its chairperson. We became a group in September 1989 at the request of Hewlett-Packard's Standards Review Board (SRB). At this time, (April 1990) we have obtained SRB approval for updates to one set of standards and have completed the first phase of the MPE/XL Standards Project.

The purpose in telling our story is to support my underlying thesis: that good communication is critical to the success of companywide migration. Good communication is required to gain an understanding of the relationships among the various people and organizations involved in the migration process. Only after understanding the multiple objectives of related organizations can we achieve smooth migration to a new technology.

I will describe the project in chronological order. Throughout the paper, I will use experiences gained from working on the project to give the reader some suggestions on how committee members can work together to achieve project objectives.

GOAL SETTING

Setting the project goals is the first and most important step in any project. The project team must have a clear understanding of what's expected of them and of what current or potential problems need to be resolved or avoided. The initial objectives of the project were outlined in the official Project Registration shown in figure 1.

PROJECT REGISTRATION
INFORMATION SYSTEMS STANDARDS

I. PROJECT TITLE: MPE/XL

II. PROJECT OBJECTIVES: STANDARDS and/or GUIDELINES for MPE/XL features for all production applications.

III. PROBLEM/BACKGROUND: Much of HP's application inventory is now running under MPE/V. Now is the time to set standards and guidelines to insure proper operation and insure efficiency as applications are migrated to this platform.

Anticiptated cost is small - 200 manhours total for a development team of 5 people.

There should be no implementation cost since it does not involve changing anything now in production. (maybe we're too late for this to be true)

Benefits will be in the consistency of usage of XL features so that the retrofit efforts like those of the last 3 years to standardize the MPE environment will not be necessary.

IV. PROJECT SCOPE: Will NOT cover migration tips - see Migration Focus Group's documentation and AEO migration guidance

Will add appropriate 3rd party tools to ISST publication

Will publish guidance for: Hardware, XL Libraries,
XL Features usage

Will recommend a STANDARD that all development teams provide dates of earliest XL support, transition period and latest MPE support date.

Will recommend STANDARD concerning the use of unique XL features in applications until all using entities are migrated.

V. DELIVERABLES: Updates to - Acct Mgmt Stds, JCL, 3rd Party Tls XL Stds will be a separate section for XL unique features which justify STANDARDS OR RECOMMENDATIONS

VI. DEVELOPMENT TEAM: (names omitted)

VII. INITIAL SCHEDULE: Registration - Sept 89
1st Review - Dec 89

Figure 1

The team's objective was to update the HP3000 application standards for MPE/XL. At the time of the project proposal, most of HP's production applications were running on MPE/V machines, and MPE/V applications had been modified to conform to the Information Technology Standards. Updating the standards for MPE/XL at this stage would ensure that both new applications and old applications migrated to MPE/XL would not require retrofitting to the standards at a later date. One downside of the early project registration was that few people in the I.T. community had experience with the new technology. As forerunners in the migration efforts, project team members had to carefully consider what impact their recommendations would have on the company as a whole.

The Project Registration was written by the Standards Review Board (SRB) sponsor for the project. Understanding the function and composition of the SRB requires some background information about the organizational structure of HP's Information Technology Community.

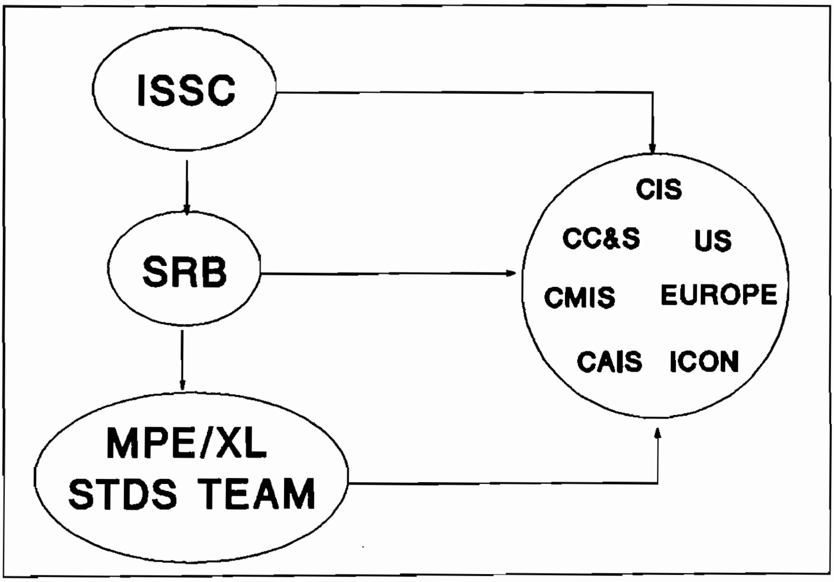
ORGANIZATIONAL STRUCTURE

Hewlett-Packard's Information Technology (or I.T.) community (formerly known as Information Systems) supports the business of HP. Our job includes selecting the most appropriate technology for business solutions. As one might expect, the HP3000, our proprietary business computer, serves as the predominant computer within I.T. Customized business applications have been developed for use internally on the HP3000 in all functional areas including marketing, administration, manufacturing, accounting, and human resources. The major applications are supported by central software groups. Each of the central software groups develops and supports applications in one functional area. Most of the central software groups are separate departments within Corporate Offices. For example, CAIS, Corporate Administrative Information Systems, supports the administrative applications that are used worldwide within HP. The internal "customers" of the central groups are the divisions that utilize their software. HP divisions rely heavily on centrally supported software for most of their business processing needs. The I.T. community comprises both corporate and divisional groups referred to as central software suppliers and receiving entities (or installers), respectively. Like HP's external customers, I.T.'s major concerns about MPE/XL and migration surround its implementation and impact on the business of our company.

The MPE/XL Standards Committee is composed of individual contributors from several different I.T. organizations. The committee was formed at the request of the Standards Review Board (SRB) which is led by Corporate Information Systems and is made up of representatives from both corporate and divisional groups. Unlike the MPE/XL Standards Committee, the SRB is composed of middle managers rather than individual contributors. The SRB is officially chartered by the Information Systems Steering Committee (ISSC) which is the governing body of the I.T. community.

Taking a top down view, the ISSC chartered the SRB to develop and maintain companywide standards, and the SRB chartered the MPE/XL Standards Committee to update the standards for MPE/XL. All three organizations include representatives from both corporate and divisional groups. On the divisional side, each organization includes representatives from Europe, the U.S., and Intercon (which represents all non-european and non-domestic HP sites). On the corporate side are representatives from central software groups and from Corporate MIS.

Organizational Structure



Acronym Key

ISSC - Information Systems Steering Committee
SRB - Standards Review Board
CIS - Corporate Information Systems
CC&S - Corporate Computing and Services (MIS)
CMIS - Corporate Manufacturing Information Systems
CAIS - Corporate Administrative Information Systems
ICON - Intercontinental

Figure 2

The SRB has authority over the I.T. standards as vested in it by the ISSC. Corporate MPE/XL Standards Team members report through the chain of command to high level managers in the ISSC (as do SRB members).

HP's original I.T. standards were developed prior to the release of the HP3000 series 900 systems and the MPE/XL operating system. In 1989, the SRB recognized the need to update the standards for MPE/XL. They chartered the MPE/XL Standards Committee whose job was to recommend updates to the standards which would then be accepted, altered, or rejected by the SRB.

My first suggestion for successful committee work is to set the objectives for the project before selecting all of the committee members. The second is to gain an overall understanding of the structure of related organizations and to obtain some historical data related to the purpose of the project. It is important to understand what organization or individual has the final say in the decisions making process. Often a committee's role is to make recommendations that influence decisions without having direct control over the final outcome.

FORMATION OF THE TEAM

Obtaining a global perspective on the migration issues was crucial to the success of the project. The project may have failed had it been attempted either by one individual or by a group that did not have proper representation from each of the major organizations. An understanding of the impact of the standards on both the software suppliers and the receiving entities would have been difficult (if not impossible) had representatives from the various groups not been included on the committee.

The initial team identified included representatives from Europe, Intercon, and Corporate. I volunteered to lead the committee and immediately solicited representation from central support teams and US divisions. We changed our name from the "MPE/XL Standards Committee" to the "MPE/XL Standards Team" and added three new members, one each from Corporate Accounting Information Systems (CAIS), Corporate Manufacturing Information Systems (CMIS), and Roseville Networks Division (RND). In addition to the team members, we identified project reviewers who would look over the documents drafted by the team members and give their input via electronic mail. The reviewers were not required to attend the meetings nor to draft any of the updates to the standards; additionally, they were not given the right to vote.

The individuals selected for the team were not only from the right organizations, they also had the necessary technical expertise. Each member had gained experience with MPE/XL by one or more of the following: alpha testing, installing hardware and/or software, participating in MPE/XL migration liaison groups, migrating applications to Native Mode, object code translating (or OCTing) compatibility mode applications, developing new applications in Native Mode, and/or teaching courses on MPE/XL migration. Moreover, each team member was wholeheartedly committed to the companywide migration effort.

My point here is to emphasize the importance of the committee selection process. A project committee should include an appropriate mix of representatives from the organizations that will be affected by the outcome of the project. It may not be necessary to obtain committee representation from each and every different group, but attempts should be made to obtain as broad a representation as possible. The individuals selected should have both a background in the technical area and a commitment to the goals of the project. For motivational reasons, team members should be informed why they were selected. This will help them to ascertain the unique contributions each can make to the project.

THE FIRST MEETING

Once the team was formed the first meeting was held where the Project Registration was carefully reviewed. The team gained a clear understanding of the expectations from the project's SRB sponsor. The project deliverables or expected results initially included updates to the official "HP3000 Account Management Standards", the "HP3000 JCL Standards", and the third party tools documents. The Team appended this list to include updates to the "HP3000 Installation Standards" and the "SL Standards for Shared Routines." Each of these documents had been written for MPE/V and served as a starting point for the project.

Also at our first meeting, the following two ground rules were established to set the foundation for the way the group would work as a team:

1) We must have at least 80% consensus from the team on all decisions.
2) All team members must make a contribution outside of the regular meetings.

It was clear from the beginning that each team member was expected to make considerable contributions towards the success of the project.

The next step was dividing up the tasks. Each of the team members was asked to review one of the current standards and to lead the effort to update that particular document. The entire team would be responsible for all recommendations, but each individual took primary ownership of one specific document. Each member volunteered to own the document that they were most interested in. The division of tasks was designed in this manner to encourage and reward the contributions of each individual and to inspire the group to work together as a team.

The following are my suggestions on conducting the team's first meeting. The leader should set the example for the team by being well prepared for the meeting. They should come to the meeting with some ideas on how to approach the project and divide the tasks, but be flexible enough to change thier approach if team members disagree. I suggest using the first meeting to motivate the group and to establish workable ground rules that encourage team participation. If at all possible, the committee should consist of individuals who have *volunteered* for the project. The leader should give members the option of leaving the team if they cannot contribute to the project. The leader should also allow each individual to select the task(s) that they are most interested in. Once the first few members select their task(s) others will naturally be motivated to volunteer for the remaining ones. The chairperson should select thier task(s) *last* and be willing to take ownership of the work considered least desirable (i.e. writing minutes.) In addition to acting as the committee chairperson, the team leader should also participate as a team member by volunteering for tasks that are roughly equivalent to those accepted by other members of the team.

WORKING TOGETHER

The team was formed at the end of September and its initial proposal to the SRB was due by the end of October. Meeting this objective required a fairly aggressive project schedule, so we agreed to meet frequently during the month of October. The schedule (shown in figure 3) was designed to encourage group discussion prior to writing up the actual standards. In addition to the formal meetings we exchanged ideas in less formal settings and via electronic mail. The formal meetings were held at Corporate Offices in Palo Alto, California which posed some logistical problems for the non-corporate team members. The representative from Intercon was only somewhat inconvenienced because his office is located in Palo Alto. The representative from Roseville Networks Division made arrangements to fly back and forth between Roseville and the San Francisco Bay Area on the scheduled meeting days. Unfortunately, for the European representative whose home office is in Geneva, Switzerland short plane flights were impossibile. He was not able to attend the meetings, so the only means of communication between him and the rest of the team was via electronic mail. The meetings were scheduled around lunch time and (except for the European representative) the team usually ate together on the meeting days. The informal lunches helped to unify the group further. As our comfort level with each other increased, so did the flow of communications and ideas.

Our second meeting was a real brainstorming session. One of the first ideas suggested was an approach to the project where migration to MPE/XL would be considered in three major categories based on the hardware options available. Applications would be categorized as being supported on either MPE/V (classic) systems only, BOTH MPE/V and MPE/XL, and MPE/XL (PA/RISC) systems only. The initial idea was to focus on developing standards for the third category and only guidelines for the second. In response, the representative from one central software group stated that his department had already established a policy of developing all new applications in Native Mode (NM). Since NM is only supported on MPE/XL, this policy implies an application classification of MPE/XL only. He commented that for his organization the second phase of support on both platforms would be relatively short. On the other hand, the representative from a different central group commented that her department had to support their applications on both platforms for a long time due to hardware constraints of the receiving entities. Clearly, the standards would have to accommodate both extremes.

MPE/XL Standards Project Schedule		
10/5	-	Brainstorm on all standards
10/12	-	First Draft of standards
10/19	-	Distribution of draft to reviewers
10/23	-	Team review of draft
10/26	-	Final draft of recommendations for SRB

Figure 3

Rather than voting on the suggested categorization early on in the project, the team decided that it was better to simply keep it in mind as they discussed other migration issues.

One suggestion to all project members is to hold off from committing to *any* ideas (yours or someone else's) too early in the project. The project schedule should include one or more brainstorming sessions where ideas are simply brought up for discussion and neither rejected nor approved. Other suggestions are to set aggressive yet realistic schedules for the projects and to participate in both formal and informal interactions with other team members.

One item that may be easily overlooked is the importance of finding ways to work around logistical issues in order to include non-local team members. Don't let transportation/location issues cause you to exclude important representatives from the team. Originally, I had asked the European team member to participate in the project only as a reviewer to simplify logistical issues. Fortunately, I was talk out of this idea by members of the Standards Review Board who informed me that this approach could jeopardize SRB approval of the team's recommendations. They suggested that the European representative participate as a full member because the European approach to standards historically differed greatly from that of the U.S. and Intercon. Excluding a European team member may have simply postponed disagreements until the issues were presented to the SRB.

When a group of people from different organizations are brought together to resolve a common problem, differing opinions will naturally surface. The team debated over *many* issues, but in the interest of brevity I will discuss only two.

DEBATE #1: MPE/XL ACCOUNTING STRUCTURE

The "HP3000 Account Management Standard" was viewed as one of the most important and well established standards for MPE/V. Much of our efforts were focused on updating this particular document which describes the components (MPE groups, users, and files) of production accounts. The standard dictates which group each type of file must reside in. It states, for example, that programs, source and job stream files must reside in groups whose names include the string "PRG", "SRC", and "JCL", respectively. As a method of updating this standard for MPE/XL, one of the first suggestions made was to add a new set of groups to production accounts for the MPE/XL versions of the files. Specifically, the new groups would contain the strings "PRGXL", "SRCXL", and "JCLXL" for NM program files, NM source files, and jobs that use new features of the MPE/XL Command Interpretter (CI). This suggestion served to initiate the discussion, but it was met with several opposing views.

The first person objecting to the idea stated that it was too great a change to the existing standards which had taken considerable time and effort to gain acceptance within the I.T. community. They suggested disallowing CM and NM files from residing in the same account and splitting the file types at the account level instead. This suggestion was also met with opposition.

More disagreements surfaced and team members were often anxious for a resolution to ensure that the October 26 deadline for the SRB submittal would be met. Team members were constantly reminded of the "80% consensus rule." Even the person who acted as the owner of a particular document, and thus did the most work to update it, was required to gain at least 80% team consensus on issues pertaining to that document.

The owner of the "HP3000 Account Management Standard" responded to the controversy by sending out questionnaires requesting votes from team members on a number of migration-related issues. These questionnaires (distributed over electronic mail) spawned more rounds of debate and allowed all team members (including the European representative) the opportunity to express their opinions and concerns.

The resolution of the account management issue was not found until later on in the project. The important point here is that we had some fundamental disagreements which could only be resolved with effective listening and cooperation. The "80% consensus rule" played a large part in enforcing effective communication. Without it, the outcome may have been based more on personality traits and politics than on what was really needed by the team and the company.

DEBATE #2: DISASTER RECOVERY

One issue that consistently came up was disaster recovery. Although we had not been asked to consider disaster recovery as part of the project, it was viewed by some as an issue that could potentially slow the migration process. Several sites were reluctant to migrate applications to MPE/XL because they only had MPE/V systems available for disaster recovery. They requested guarantees from the central software suppliers regarding backward compatibility to MPE/V (note that HP traditionally supports forward, but not backward, compatibility). If backwards compatibility from MPE/XL to MPE/V were *always* a requirement, then development teams would be prohibited from using *any* new features of MPE/XL that are not supported on MPE/V. This would prohibit the use of Native Mode, new CI features, and command files until the entire company had migrated to MPE/XL. The replacement of all MPE/V systems with MPE/XL systems within HP's I.T. community is expected to take several years. To some central software teams, it didn't make sense to wait this long before taking advantage of the new features of MPE/XL. On the other hand, some divisions were adamant about backward compatibility to MPE/V.

The team discussions about disaster recovery were particularly heated and involved. Several claimed that disaster recovery should not be a topic for MPE/XL standards while others argued that ignoring it would impede companywide migration efforts. The responsibility for disaster recovery belongs to each individual HP site. In general, divisions make reciprocal agreements with each other to act as backup sites in the case of a disaster. One of the corporate MPE/XL Standards Team members had previously assisted a division in moving an application they supported to MPE/XL. This site had a disaster recovery plan in place that was based on MPE/V systems for recovery. At that time, HPPA systems were *not* commonplace and it was difficult to find a site with MPE/XL systems for a reciprocal disaster recovery agreement. This site *insisted* that the central software supplier guarantee backwards compatibility to MPE/V before they would move their production application to MPE/XL. For performance and MPE/V compatibility, the application was run in object code translated Compatibility Mode (or OCT'd CM). Without getting too technical, I will simply state that this particular application could not be ported back to MPE/V with a simple STORE/RESTORE. After OCTing, the application's SL was larger than one extent and the MPE/V loader would fail when attempts were made to execute programs that accessed it. To work around this problem, the production environment for this site was altered to include both a CM and an OCT'd CM version of the application: the OCT'd CM version for better performance during regular production and the non-OCT'd version in case recovery was needed on an MPE/V system.

Was this problem just a fluke that could be ignored? Was it really a subject for MPE/XL standards? Why couldn't we insist on a like- platform disaster recovery policy? The debate went on and on; at times it seemed that there was no end in sight. The team could only agree to disagree and time was running short. As with the accounting structure debate, the consensus rule forced cooperation eventually leading to a resolution (to be explained). My point here is to stress that it's okay to disagree. An added suggestion is to find out what real-life experience is the basis for the disagreement.

BREAKTHROUGH

It took the concerted effort of the entire team to gain a thorough understanding of the issues and to resolve them with well-defined classifications. We finally came to an agreement at one memorable meeting where hard work, determination, and the willingness to listen to opposing points of view resulted in a communications breakthrough for the team. (This meeting was followed by an enjoyable and well-deserved lunch break!)

The solution is based on one of the first ideas expressed at the initial brainstorming session: a classification of applications based on the supported hardware platform(s). At the breakthrough meeting, we realized that the issues had been muddled in our individual assumptions about the meaning of terms. One team member wrote the three classifications on the board and we all debated over the precise definitions and implications. We finally came to a consensus on the three classifications (shown in figure 4) of "MPE/V", "MPE/V & XL", and "MPE/XL".

The classification of "MPE/V" only is used for nonstrategic applications or those that will be replaced by newer applications within the next few years. The middle classification of "MPE/V & XL" is for application versions supported on MPE/V and in CM (or OCT'd CM) on MPE/XL. The third classification of "MPE/XL" is for any application using a new feature of MPE/XL that is not supported on MPE/V. During the transition stage when an application is needed on both MPE/V and MPE/XL systems, the central supplier of the application has two options: either support one application version on both hardware platforms ("MPE/V & XL") or support two distinct versions on each of the platforms ("MPE/V" and "MPE/XL"). The first option simplifies support issues, but prohibits the supplier from using new features of MPE/XL. The second enables the development teams to use new MPE/XL features on the "MPE/XL" version while still providing support for an "MPE/V" version for the sites that have not yet migrated.

The beauty of the classifications at the application version level is in its clarification of complex issues. Simply communicating the supported hardware platform(s) for each application resolves many of the migration issues.

For the disaster recovery issue, there is no ambiguity for any application classified in the "MPE/XL" category, they *must* recovered on an MPE/XL system. For applications classified as "MPE/V & XL", the team added the stipulation that central suppliers be required to document any special procedures needed to recover the application from an MPE/XL system to an MPE/V system. This covered the special case where OCTing led to backwards incompatibility.

Excerpt from HP3000
Account Management Standards

2.2.1 Application Classification for the Supported Hardware and Software Environment

In addition to the classifications defined for security, specific versions of an application will fall into one of the following categories:

MPE/V : Assigned to each application version which is supported ONLY on MPE/V systems.

MPE/V & XL: Assigned to each application version which is supported and will run on either machine. The software supplier must provide a means of allowing the application to run on both machines. New applications should not be created in this category. The intent of this category is to be able to move an application from MPE/XL to MPE/V when necessary.

MPE/XL : Assigned to each application version which is supported ONLY on HP3000 900 series machines. If an applications version must be recovered on an MPE/XL system, then it must be classified "MPE/XL" only.

It is required that every application be assigned one of the above hardware/software support classifications. This classification is assigned by the software supplier. The application is supported and must be able to run on the machines indicated by the assigned category. The MPE/V & XL category is intended for those applications which have been ported from MPE/V to MPE/XL and have been OCTCOMPed. If there is no intention for the application to return to MPE/V, then it should be classified as MPE/XL.

Figure 4

The accounting structure issues were also simplified by classifications at the application level because it allows the MPE/V accounting structure to be used. Only CM files reside in the groups for applications classified as "MPE/V" and "MPE/V & XL"; whereas only NM files reside in the groups for "MPE/XL" applications.

Consider a sample distributed application currently used in production. Should it be supported only in CM, converted to NM, OCT'd, or supported only on MPE/V? The answer to this question depends upon the support resources of the central supplier, the hardware resources of the receiving entities, and the longevity of the application. If there are no receiving entities who require migration to MPE/XL, then the application should be classified as "MPE/V" to avoid unnecessary costs associated with migration and support on two hardware platforms. On the other hand, if there are receiving entities who require support on the series 900 systems, then the application should be classified as either "MPE/V & XL" or "MPE/XL". Selection between the later two depends on whether or not new MPE/XL features are utilized by the application. Whichever option is selected, it is imperative that the supplier communicate the intended hardware platforms to ensure that their plans are in synch with those of the receiving entities.

Communication is the key to simplifying the migration process. The new classification requirements force the central suppliers to clearly communicate their support plans with respect to MPE/XL. This allows the receiving entities to plan accordingly.

The chaos of unanswered questions was cleared with a solution that now seems obvious. The solution, based on clear communication between software suppliers and installers, was attained through good communication of MPE/XL Standards Team members.

The communications breakthrough occurred at our meeting on October 12th.

... AND THEN

On October 17, 1989, the San Francisco Bay Area experienced a major earthquake. It registered at 7.0 on the Richter scale and lasted a total of 15 seconds.

Suddenly, the MPE/XL standards project became insignificant. We all survived, but we were all affected. Our concerns quickly shifted to worries about the health and whereabouts of our families and friends. The news was filled with stories of destruction, death, heroism, sorrow, and sometimes even joy. The ground had stopped shaking, but reminders of the event surrounded us. Our minds were filled with questions, ones that trivialized our concerns about MPE/XL migration.

No one seemed to notice that the project schedule had slipped. The scheduled distribution of drafts to reviewers did not occur on October 19th.

And yet somehow ... life goes on.

In projects (as in life) we must learn to deal with the unexpected. "Disasters" of any size may interrupt our progress, but they should not cause us to abandon all objectives. We must somehow find the strength to proceed. Nevertheless, I believe it is wrong to deny the emotional effects of a disaster. The "strength to proceed" may only be obtained after a period of mourning.

Fortunately, HP management did not enforce an unreasonable policy of "business as usual" immediately following the earthquake. Their actions, which speak louder than words, conveyed compassion and concern for employees. In fact, individual contributors and managers alike expressed emotional support for each other after the devastating event. The MPE/XL Standards Team was no exception.

At our first meeting after the earthquake, I brought in a rose from my garden. This gesture was intended to counteract the effects of the missing ceiling tiles and long cracks in the stairway walls. People require a comfortable physical environment in order to be productive, but in this case the environment had been damaged: unnerving evidence of the quake was everywhere.

The synergy of the MPE/XL Standards Team proved to be far more powerful than either the symbolic rose or the damaged physical environment. It is difficult to explained what happened at our meeting on October 23rd. We started it off with stories about where we were when the earthquake hit and moved into sympathetic discussions about those who suffered and our feeling about living in the Bay Area. We supported each other through a difficult time. Together we gained the strength needed to continue the project.

What "words of wisdom" can I say? How about "be human?" You may never experience an earthquake, but you may experience human tragedies, losses, and unexpected events in the middle of a project. Your method of coping with such events is most likely different from mine, but don't deny yourself (nor others) the opportunity to to express pain, sorrow, and other negative emotions in the aftermath. Allow for a period of mourning over the loss, give and receive support from each other, and *then* pick up the pieces and move on. Also, if a tragedy occurs do little things to help alleviate the tension. I used a rose to decorate and lighten up the environment, but you should use whatever seems most appropriate that will temporarily take everyone's mind off the event.

COMPLETION

Even with the earthquake (or perhaps because of it) the team managed to complete the first phase of the project. At our meeting on October 23rd, we discussed what could realistically be achieved by the due date, which was October 26. We wanted to come up with some recommendations that the SRB could vote on and either approve, alter, or reject.

Our first recommendation was the new requirement for applications to be classified with respect to the supported hardware platform(s). The second was the date for compliance to this new standard. The third recommendation was approval of the standards as updated by the team. The updated versions were included in a package that was sent with a cover letter (excerpt in figure 5) to the chairman of the SRB. Our last request to the SRB was for guidance on the best approach for an update or a new standard with respect to the introduction of the LinkEditor. Two team members had taken ownership of the document titled "SL Standards for Shared Routines" and they each had very differing views on how it should be updated. The recommendations of both team members were included in the package to the SRB with a request that the SRB make a selection between them.

Excerpt of Cover Letter of the
MPE/XL Standards Committee Proposal

Our proposal to the Standards Review Board is as follows:

1. All centrally supplied application versions must be classified as either MPE/V, MPE/V & XL, or MPE/XL as defined in the draft of HP3000 Account Management Standards Version 1.5 section 2.2.0.

2. The software suppliers must classify their applications by March 17,1990.

3. We recommend updates to the following current standards:

   (1) HP3000 Account Management Standards
   (2) HP3000 JCL Standards
   (3) HP3000 Installation Standards

4. We also recommend the addition of a standard document for third party tools. The information collected by this committee and documented in "Third Party Software Tools for MPE/XL" should be used to produce the new standard.

Figure 5

My advice to you here is: don't wait until the project is absolutely finished to recognize the progress made at specific points in time and to report that progress back to the individual or organization that chartered the project. Also, if the committee isn't able to agree on an issue prior to a due date, own up to that fact and request guidance from the group (or individual) who set the date. Decision maker(s) need to be made aware of issues that impede progress. If more time is needed to complete the project, the team should request more time.

CELEBRATION

On the evening before the SRB meeting, we went out to a local restaurant to celebrate completion of the proposal. We had refreshments, some good food, and time to simply relax and enjoy each others' company. The celebration was courtesy of our SRB sponsor.

All successful team projects should include similar celebrations.

THE SRB MEETING

MPE/XL standards was the first item on the agenda of the December SRB meeting. When presented, the first two items of the proposal were unanimously approved! Some of the details in the standards documents provoked discussions among SRB members that were very similar to the earlier ones among the MPE/XL Standards Team. We responded to their questions, indicating that their concerns had been addressed and we negotiated the wording in several of the documents. In the end, the SRB agreed with our basic recommendations and *approved* the first set of standards. They even gave us a round of applause for a job well done!

CONCLUSIONS

I've alluded to the fact that the project is not finished. We have completed our *first* major milestone and gained approval of one set of standards, but there is still more work to be done. The team is currently working on a document titled "Standards for Shared Routines" which describes how code in external libraries (SL's on MPE/V and XL's on MPE/XL) is shared among development groups. The LinkEditor on MPE/XL presents us with more options (and hence more standards challenges) than its predecessor technology, the Segmenter on MPE/XL. I am confident that what we learned in the first phase of the project will guide us towards successful completion of the next.

What we have learned is summarized in figure 6 and my underlying thesis. To reiterate, good communication is critical to the success of companywide migration. It's important to listen to the concerns of individuals in organizations other than your own and to recognize the validity of their concerns. In this fast paced world of ever changing new technologies, it's tempting to push forward in the direction we're most comfortable with and ignore any differing points of view. We must take the time to gain an understanding of the people and the organizations affected by the introduction of a new technology. In short, we must learn to respect the opinions of others. An understanding of people and the organizations they represent can only be achieved through good communication.

The technical challenges presented by migration to MPE/XL are minimal compared to the human and organizational challenges. Of course, through hard work, determination, and a commitment to *good communication* any challenge is surmountable!

Tips for Effective Team Management

1. Set objectives prior to team selection.
2. Understand the organizational structure.
3. Know the history leading up to the project.
4. Know who has the final say in the decision making process.
5. Select an appropriate mix of representatives.
6. Select individuals with the right technical expertise.
7. Select people committed to project objectives.
8. Establish ground rules that encourage participation.
9. Make participation voluntary.
10. Allow latitude in task selection and approach.
11. Include brainstorming sessions in the project schedule.
12. Don't commit to ideas too early.
13. Set an aggressive yet realistic project schedule.
14. Work out logistical issues for non-local team membership.
15. Interact both formally and informally.
16. Expect and allow for disagreements/debates.
17. Find out if disagreements are based on experience.
18. Use a consensus rule.
19. Be human. If a tragedy occurs, allow mourning.
20. Support each other.
21. Alleviate tension with simple gestures.
22. Recognize completion of phases.
23. Request guidance if needed.
24. Celebrate success!

Figure 6

Maintaining A Quality Staff

Forrest Ford
Forrest Ford Consultants, Inc.
680 Craig Road, Suite 105
St. Louis, MO 63141
(314) 567-6737

Introduction

When creating a staff in the information systems industry, managers look for technical expertise as well as applications and industry experience. Creating an exceptional staff involves the added dimensions of interpersonal skills and the ability to work as a team. The ultimate goal of these teambuilding efforts is maximizing profit, the bottom line of business. Once the elements of a quality staff are in place, maintenance is critical and best accomplished by cultivating a group of employees that is committed to achieving the goals of the corporation in conjunction with their own.

Customer Service

The most important goal of any business is caring for the client. The customer service mentality understands the customer always comes first consistently and constantly. It involves giving the client what he wants in a quality fashion in the timeframe requested. Since a business' longevity is so closely tied to its clients' patronage, all activity must somehow be connected to satisfying the needs of this corporate lifeline.

How does this translate to the information systems industry? Consultants are generally viewed by the client as "experts" called in to rectify a problem situation. The "expert" mentality is schooled and geared toward finding the most technically sound solutions possible. Conflict arises when solutions, although textbook-perfect, are not functional given the work environment and personalities involved.

The most service-oriented consultants give the customer what is requested, not necessarily what is needed. The consultant's responsibility to the client is to provide as many viable solutions as possible, honor the client's privilege to choose the option that seems most appropriate, and support the efforts toward the stated goal. It is important to remember that the resources involved in this process belong to the client. The consultant is merely a guest invited to advise the customer on how to best use these resources.

Product Quality

Producing an outstanding product is the second most important corporate goal. Producing a product that equals or exceeds industry standards reflects the attitude, "We can do it and we are glad you gave us the opportunity!" Serving the customer means providing a product that "solves the problem" without bias to hardware, methodology, or philosophy. Achieving this level of service is nearly impossible without teamwork.

The Importance of Teamwork

Teamwork provides a variety of opportunities to participants. It helps insure effectiveness, that is, making sure that the job completed is the **right** one. It tests employees' ability to work in difficult situations with difficult people. It provides a pool of talent from which to draw in completing assignments.

It also involves joint accountability. This keeps individual efforts directed toward the common goal producing a synergistic effect toward overall progress. It is based on the fact that everyone is part of the company team. Each employee shares the group success and takes responsibility for the failures. Joint accountability is also based on commitment; you can't dabble in it. With "everybody in the same boat," it is important to remember that you cannot sink half a ship, nor can you keep half afloat. Such replies from employees as, "it's not my job," undermine the group effort and stall progress. By undertaking duties not covered in their job descriptions, employees fill an important gap in the process and gain the opportunity to explore other areas.

Employee Growth: Personal and Professional

With customer service and product quality established as the corporate goals, the basics for survival, no other priority is more important than the quality of employees. Commitment to the personal and professional growth of employees is a commitment to the company's most valuable assets and vital to building a quality operation.

A manager is the primary caretaker of these company assets. The most important function of a manager is to coach, teach, and inspire employees to be motivated to do the great things they are capable of doing. To inspire employees to reach their potential, a manager must first be in control of himself.

Happiness = Motivation = Productivity

Psychologist Dr. Jane Brownstone advocates, "The more we do for ourselves regarding our emotional development and happiness, the more we are able to do for others. While it sounds selfish, the theory in practice allows us to be in better control of our own happiness, putting us in a better position to assist and direct others, or for that matter, direct any endeavor."

Working each day to be happier with ourselves, is a practical, constructive way to inspire others through example. Dr. Abraham Maslow theorizes, "Most people become a little less happy each year of their lives. They forget or fail to practice being happy and taking care of themselves is the necessary ways such as diet and exercise." Dr. Maslow believes, "People should work each day to inject a little more happiness into their lives than the day before. If practiced faithfully, the theory follows that each day will be increasingly happier with the last day of life being the happiest of all."

How do we motivate ourselves, and therefore inspire others, to take the action necessary to make our lives happier, more productive for ourselves and for society? Author Joseph Campbell encourages society to "Follow Your Bliss!" To do this, we must follow the calling and directives we "hear" inside ourselves, even when they conflict with the paths others have chosen for us. For example, if a child's parents plan for him to be a lawyer or doctor when he grows up and he dreams of being a fire fighter, Campbell says, "By all means, go fight fires! Follow you bliss, don't spend your time following someone else's bliss. That would be wasting your time, wasting you life." 1

Toward this end, how do we put more happiness, that is the positive feeling of fulfillment, into our lives. We begin by respecting ourselves and our roots.
In Guidepost Magazine, Dr. Norman Vincent Peale said, "If we have 300 billion brothers and sisters, we would still be unique. We are special and there will never another person like us." This theory holds a variety of messages for us.

Fulfillment Begins with Respect

We should never feel inferior to anyone else. We are equally unique, therefore, no one can be more or less worthwhile that anyone else. We can never grow up to be someone important . . . we already are! We should never be ashamed of our parents, family, or background. They are important components of our person. Feeling secure and satisfied about the elements connected to our make-up helps us feel secure and satisfied about ourselves.

Because we are unique, we should consider the possibility that we also have a purpose, a responsibility to fulfill, a reason for being. We should dedicate ourselves to discovering its meaning and developing to our fullest potential. If we fail in this action, the goal will not be completed. Our special, unique approach to the task cannot be duplicated by another person; our destiny cannot be fulfilled by another; it is ours alone.

With self-respect as a foundation, greater happiness and greater productivity can be achieved by adopting the following principles. Only by adopting them ourselves can we inspire others to adopt them.

▪ Be willing to take risks and expect to make mistakes. People who do not make mistakes do not do anything and therefore never get anywhere. Progress cannot be made by staying safe.

▪ Dwell on the positive and refuse to be defeated by the negative. This takes focus and faith, but without them, many great contributions would never have been made. Dr. Seuss visited and was rejected by 23 publishers before The Cat in the Hat was published by the 24th.

▪ Do not criticize others negatively. It demonstrates small-mindedness and defies the "Golden Rule." According to author Frank Clark, "Gossip need not be false to be evil. There's a lot of truth that shouldn't be passed around."

▪ Develop a sense of urgency about our work and our lives. In his lectures, psychologist Harles Cone encourages people to, "Ready-aim-fire! It is easier to guide something once it gets going than it is to get it started in the first place." Like a guided missile, circumstances and instincts will eventually direct us to our goal. In the words of Will Rogers, "Even if you are on the right track, you'll get run over if you just sit there!"

■ Employ the Make Me Feel Important (MMFI) Principle. This is the most intense non-biological need humans experience. Therefore, in dealing with others, work to make them feel worthwhile and communicate the importance of their contributions.

■ Encourage yourself to follow your intuition. "Intuition is always 100% right, but it takes time to *hear* it correctly. If you are willing to risk acting on what you believe to be true, and risk making mistakes, you will learn very fast by paying attention to what works and what doesn't work. If you hold back out of fear of being wrong, learning to trust your intuition could take a lifetime." 2

■ Use your own best judgement. The Seattle-based retail conglomerate, Nordstrom Corporation, illustrates this in the policy statement to its employees. Their message is:

1. We trust you, so trust yourself.
2. We know your best judgement represents us well.
3. We are all part of the same successful team.
4. What you think and how you act are important to us.

The value of having faith in yourself is proved by Nordstrom's overwhelming, long-term success.

Case Study

With any theory, the value is difficult to determine until it has been properly tested. One case where this method of employee commitment has been extremely successful involves an information systems consulting firm. From its inception twenty years ago, customer service and product quality have been the primary company objectives. Beyond that, nothing has been more important than the employee. This commitment is outlined in what has become the "Fifty/Fifty Business Relationship," an offer extended by the president to each of the company's 250 employees.

In this agreement, the president pledges his commitment to the employee's needs and development. The time the employee invests in the company is highly appreciated, since these eight hours per day can never be retrieved. In return, the employee is asked to commit to the company goals and leadership. The president guarantees 50% of the relationship. If the employee offers another 50%, the relationship is an ideal 100% commitment. If the employee is interested in commiting 20%, the total will only be 70%; while it is still a working relationship, the employee has no grounds for complaint if the career does not develop to satisfaction.

This policy illustrates the company's commitment to the employees' personal needs as long as they are equal or subordinate to the corporate goals. The successful results illustrate the policy's value. This "partnership" also makes employees' success as much their responsibility as the company's, giving them a sense of control about their progress and development potential as well as a place to begin evaluating when expectations fall short.

In addition to the commitment the company asks of its employees, it asks an even greater commitment of the employee to the customer as reflected in the following passage paraphrased from an essay by Elbert Hubbard.

> If you work for a customer, in heaven's name **work** for him. If he pays you the wages that supply your bread & butter, stand by him, and stand by the institution he represents.
>
> If put to the test, an ounce of loyalty is worth a pound of cleverness, but if you must vilify, disparage and eternally condemn, then resign your position.
>
> And when you are outside, you can damn to your heart's content, but as long as you are part of the customer's organization, do not condemn it, for if you do, you are loosening the tendrils that hold you to that organization and the first strong wind that comes along will blow you away and you will probably never know why.

Conclusion

In the final analysis, we all belong to the same professional community. Because of this, the ultimate goal is to care for our customers by providing exceptional service rather than to concentrate energies on defeating the competition. The successful attainment of these corporate goals is dependent on the extent to which the staff is committed to them. Time and energy spent cultivating employees personally and professionally will be realized in the service quality and client satisfaction. The companies that endure are the ones that understand the value of doing business in this constructive manner.

References

(1) Campbell, Joseph, "*The Power of Myth*," (New York, Doubleday, 1988)

(2) Gawain, Shakti, "*Living in the Light*," (San Rafael, CA, New World Library, 1986)

TITLE: Migrating Compatibility Mode Systems
to Native Mode

AUTHOR: Kevin Darling
The Gap, Inc.
3434 Mineola Pike
Erlanger, KY 41018
606-283-1944

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 5085

Project Management - Too Simple to Believe

David J. Rubinstein
Vice President
Innovative Information Systems Inc.
63 Nahatan Street
Norwood, MA. 02062
617-769-7511

Through the experiences I have had in data processing departments and as a Systems Integrator, the most common weakness I have seen in the development life cycle is Project Management (PM). PM is what I would term the poor cousin in most systems development environments. I have chosen to address this topic because I feel this is one area which would provide the greatest 'bang for the buck' in systems development.

Why is it that PM is not treated as an asset by many developers? Some companies I have worked with are excellent at project management. In fact, many companies, such as AT&T and Mcdonnell Douglas, are the major contributors to project management and system development methodologies. They have taken the time and have had the foresight to plan ahead. Organizations such as these understand the value of long-term planning.

Several reasons why I believe project management is a neglected tool are fear of complexity, overhead, misunderstanding, and short-term orientation. In fact, I believe that it is this 'short-term' orientation that brings about most of these problems. American business is totally enamored with short-term results. On the macro level, LBO's, junk bonds, and stock holder pressure for short-term profits are our greatest examples of our shortsightedness. Pressures such as these force management to demand results from MIS and other business units. Because of this orientation, we neglect valuable tools such as project management.

The purpose of this paper is to discuss the benefits of project management. What I hope to convey is a simple, common-sense understanding and approach to managing our projects. If you are already using project management techniques and your company is committed to this practice, then this paper is not for you. However, if you are trying to implement project management, attempting to convince your management that it is a necessity, or have an interest in using these tools, please continue on.

Yes, project management will take up some of your time and it will cost money. However, a project which is ill-planned and ill-controlled will go over budget by 4X versus a well planned and controlled project (which may go over budget only 2X). Many of us view project management as overhead, I view it as not only as a necessity, but a value-added service to any project.

WHAT IS PROJECT MANAGEMENT.

A project is any irregular undertaking performed by people to accomplish a defined set of objectives. Included in these objectives are quality, time frame, and budget. According to Webster, a project is a planned undertaking. What this all boils down to is the need to plan an approach in order to accomplish our defined objectives. Not only must the path to success be planned, it must be managed.

An analogy to this would be a vacation. Lets say you were to go sightseeing from Los Angeles to San Francisco. When you plan a vacation, typically you start with a budget, how much can I spend. Secondly, you have a schedule to meet, maybe one or two weeks of time. Lastly, you have an objective to have fun, relax, sightsee, etc. In order to maximize your final objective, you have to plan the first two items. To start, you know what you want to spend, how long you will be on vacation, and what you would like to do. Throughout the course of the vacation, you have to monitor your spending, you must plan what city or town you will stay in next, and you must ensure that you are fulfilling the objectives of your adventure. You are essentially the project manager.

The role of project management is the most critical ingredient in your project (or vacation) success. Project management is the process of planning and controlling a project. As we have seen in the vacation example, the project manager has a great deal of impact on the success of the project. In fact, without the project manager, there is no one to allocate resources and plan ahead.

In our development environment, it is the project manager which must maintain this delicate balance. It is project management that creates the balance between people, the work which must be accomplished, and the technology which is utilized.

WHERE TO START

Until now, we have been justifying the importance of project management. What has been established is project management is essential to the creation and on-going health and success of a project. In order to have project management, one or more people must act as the project manager(s). For a project to begin, there must exist an objective or statement of purpose.

A. The Project Objective.

For a project to exist, an objective must be stated. In many organizations, projects come and projects go with no stated purpose. Many times its "cause the user asked for it." If there is no stated objective, how do we know when the project has ended. Many times the project does not end, and management looks at MIS as says "you're still working on that thing, when will it be over".

Project management helps to eliminate that perception. By stating an objective, we begin to create the building blocks of a project. We can define the start and end of a project. If the stated objective is "to process customer orders in under 45 seconds" or "reconcile sale taxes for audit purposes", then we have a defined objective. Have the business user participate in defining the objectives. Keep away from computer-eze. One of the key ingredients is user/management commitment and participation in our projects.

This brings out another key ingredient in the planning process, management commitment. As one of my clients once told me, he does not want management involvement, he wants commitment. He wants them to participate in the project objectives, the planning process, and be part of the project work. Involvement is to tangential.

He once told me the story of having eggs and bacon for breakfast. The chicken is involved, because it gave us the eggs for breakfast. However, it is the pig which is truly committed. Both my client and I believe strongly that management commitment is a key to project success.

As we will see (and as you have probably experienced), management can put a strain on our project resources, through scope expansion and diversion of resources. Management commitment can help educate management what is needed in order to complete a project. Also, through their commitment, they will take pride in the project deliverables.

B. Creating the Plan.

Now that we have established our objectives, we need to create a project plan. The creation of a project workplan is a detailed, tedious task. It is absolutely necessary. How could we go on our vacation if we do not book airplane reservations, reserve a rental car, and reserve hotel rooms. The same concepts apply to a project. How can we process customer orders without defining functional and technical requirements.

Our development projects are never simple. We may perceive them as such, but they should not be treated in this manner. What we are trying to do is solve a complex problem by breaking it up into a series of simple tasks which lead us down a path.

As part of project planning we must establish what these tasks are and how we will go about completing them. There are many published methodologies which can act as a guide to creating project plans.

In creating these tasks, there will be a sequence of events. Many of these tasks will be dependent on one or many preceding tasks. Part of project management is to establish these dependencies. This is what helps to establish the 'critical path'. The critical path consists of a series of tasks which have the greatest bearing on the project timetable and budget.

Lastly, do not forget the tasks which require management commitment. There are typically tasks which require management signoff, such as approval of functional requirements. Also, quality assurance checkpoints throughout the project guarranty that the business understands what they will be receiving and that they approve of the product.

C. Estimating.

The project tasks are now in place. The next step is to get an idea of how long each task will take. Unfortunately, in software development there is not set formula for estimating tasks. Estimating is an acquired taste (like drinking Scotch). We can work within a reasonable range. One rule I have is that nothing is coded and tested in less than 8 hours, even the simplest report.

The purpose of estimating is to establish what the worst case is without compromising the project objectives. You want to establish the 'not to exceed' scenario. By creating a conservative estimate, you are establishing a more realistic timetable and budget. Most estimators tend to be too optimistic. I have not seen many estimates come in under budget.

We run across many overly optimistic people. There is a belief of "I can bring more people on the project". Typically, the later you bring people onto a project, the later the project is completed. Some other concerns we need to keep in mind are: vacations, sick days, training time, holidays, and other human type factors. I was once on a project that had the team members budgeted at 60 hours a week. Needless to say, we did not accomplish any of our objectives. However, we were very burnt out after nine months and poorly productive. So don't think that overtime is a cure to project productivity either.

D. The Schedule.

We now have our tasks and our estimates, therefore the schedule can be created. A schedule is the sequence of events which lead to satisfying our defined objectives. The schedule takes into account estimates, identified resources, and other parameters.

In creating the schedule, we will be working with many dependencies among tasks and resources. This will lead to the critical path (mentioned earlier) and also critical resources. We may find that we do not have enough programmers, or that a code generator may be required. Not only is scheduling of tasks critical, so is the scheduling of resources.

The schedule will be a constantly changing animal throughout the project. It is essential that the schedule is communicated to project participants on a timely basis. We like to review the schedule and estimates with all project team members. Their feedback is essential in establishing whether our goals can be achieved in a realistic time frame.

E. Control.

Throughout this process, we have been reviewing the steps project management must take in order to organize and plan the project. At this point we are ready to being the project. However, our job as project managers is not complete.

As project managers, we must now control the project. Until now we have created the plan. As part of this plan, we need to implement mechanisms which allow us to monitor the project.

We have create a schedule with estimated resources and times. We must track our actual expenditure of these resources versus the plan. On our vacation, if we get to Big Sur on day 4 of 7, we will have to adjust our plans in order to get to San Francisco. We either extend our vacation, or we sacrifice without compromising our objectives. The same must be dealt with in our development projects.

Management may have asked for a scope change or for a programmer to work on something else unrelated to our project, but critical to the business. This will require a reshuffling of our time frame, resources, etc. It important that these concepts are related to management. They will gain an understanding of what their requirements can do to the project, and also realize what it means to "rob Peter in order to pay Paul".

What management has just done has caused us to take corrective action. We must now somehow compensate for the change in resources. Monitoring and controlling the project is a necessity. No project exists without taking corrective action.

In order to be able to take the proper action, we need to monitor on an on-going basis. This requires creating feedback mechanisms. In many instances, a time report sheet is utilized to track how much time has been expended on an individual task. These times are then tracked against the estimates. We also ask team members to provide an estimate to complete on their tasks. This may change the original estimate, which also impacts the schedule.

Along with time sheets, we like to use a narrative project progress worksheet. we ask for narratives because the project story does not just exist in numbers. We want to have a comprehensive understanding about all of the issues involved, not just the numbers. This should also be extended to project team meetings.

In order to reestablish estimates and schedules, a project management system must be utilized. It can be manual, a spreadsheet or an elaborate system such a Harvard Project Manager or Project Manager's Workbench. Whatever the tool being used, progress reporting and feedback is essential.

Actual resources (time) should be tracked against estimated resources. This will tell us a great deal about project progress. We will be able to better identify potential bottlenecks. This is extremely useful in monitoring the critical path and critical resources. If the entire system is coded, tested and ready for production, but the users have not been trained, then a serious flaw has been made in the project.

Having control over the project and managing its resources appropriately can lead to success alot easier than without it. However, you must be committed to these ideas. Just giving lip service will not get it done.

Will project management cost me anything, Yes. But it is cheaper to complete a project with it than without it. It is easier and more likely that you will fulfill your objectives, including quality, budget, and time frame, with it than without it.

TITLE: The Greeting Behavior of the North American Homo Sapiens

AUTHOR: Louis R. Mills
Bio-Rad Laboratories
1000 Alfred Nobel Drive
Hercules, CA 94547
415-724-5613

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 5124

"TINKER-WARE": MAXIMIZING YOUR INVESTMENT IN APPLICATIONS TOOLS

Leonard Block
The Apex Group
7151 Columbia Gateway Drive, Suite F
Columbia, Maryland 21046
(301) 290-1606

INTRODUCTION

The best way to describe "Tinker-Ware" is for me to take you inside my children's toy room (ages 6 and 3). They have hundreds of parts (the mess makes it appear so) from a variety of toys all over the place. When I watch them play, they don't play with just one complete toy, but rather a piece from here and there and there etc. Sometimes, simple things like our children's playing habits can teach us principals that we can use in our every day life. The applications tools that we use in our data processing shops can also be assembled in a variety of ways to build new and improved processes. Like our children's tinker toys to build new playthings, we have what I call "Tinker - Ware" to build new production processes.

The applications development tools available today for the HP3000 keep expanding day by day. A great variety of packages and products can be obtained for your tool box from the following areas: applications software packages, data base management systems and utilities traditional and 4GL software languages, word processing, spreadsheets, graphics, text editing, data communications etc. The list can keep going on and on.

Even without buying additional tools, practical processes and routines can be built with software that comes standard with your HP3000 that one might not be aware of. These new processes might be limited in some of their capabilities but remember you get what you pay for. If you need more capabilities, then go out and evaluate third party software on a cost benefit basis. The point being, check your own backyard before spending some additional money.

When one does purchase additional products, companies often do not realize how to maximize the investment they are making in the product. Purchases should be looked at in the context of what other tools exist and how the new member of the family will fit. Maybe it was not needed at all if two existing tools were combined with a little programming support. The goal of this paper is to create an awareness of how one can better utilize their applications tools without necessarily buying any new software. We will first look at how our applications tool box has evolved and what one should consider when building one for their department. After that, some practical ways to integrate application tools along with some detail case studies of successful "Tinker-Ware" projects. Please note that this paper is not intended to endorse any particular vendor or product. The examples used are based on application tools that were available in the tool box of the company at the time of integration. Let's begin the journey.

TOOL BOX EVOLUTION

In the beginning.......... HP created the 3000 series and a whole new family of business computers were born. Like our history, the tools available with the HP3000 series have progressed and expanded through different eras. Each era building upon the previous one. Today things are changing so rapidly that new options are becoming available for your HP3000 as you read this. Lets take a brief run through this tool box evolution.

Stone Age

The Stone Age is characterized by the traditional tools that one either got as part of the purchase of an HP3000 or were common in the industry at the time. Pre HP3000 data processing will not be addressed for this discussion. Vendor supplied software and utilities were often the only tools companies could work with. IMAGE, QUERY, SPL, SPOOK, FCOPY, EDITOR along with COBOL, BASIC & FORTRAN (3rd generation) compilers were in a nutshell the applications tool box for programmers. Applications software that one could buy off the shelf (payroll, G/L, etc.) was also very limited. With all our choices today, those times did indeed seem somewhat prehistoric.

Renaissance

As the HP3000 evolved, so did the available tools to be used in database management systems, file utilities, text editing, high level programming languages as well as enhanced applications software packages for companies. I like to call this the Renaissance period because it represents an awakening. Third parties started to realize the tremendous market potential that the 3000 line of computers represented. An interactive oriented mini computer with easy upward compatibility in its operating system, geared towards business solutions seemed a cost effective alternative for many companies. The Renaissance period enhanced our applications tool box in three major areas. The first is third party applications packages. As mentioned, this growing market needed good payroll, financial and manufacturing software. Third party vendors started to become major players in replacing homegrown written systems developed during the Stone Age. Second, the vendor supplied utilities which came with the system started to be improved upon by third parties. Faster more powerful, full screen editors started to appear on the market. File utilities, database management tools, job schedulers, graphics, security packages, faster compilers, data extraction/sort and numerous other areas were being offered to help expand a company's applications tool box. These tools worked very well with existing HP tools and file structures. Not only did third parties develop improved applications tools, but HP itself kept working to improve on its original tool box.

The last major trend in the Renaissance period is the advent of 4th generation languages. The user friendly nature of the HP makes it a perfect marriage for the characteristics of the 4th generation languages as compared to a batch oriented processing environment. In this book of proceedings, there are probably several papers on the concept of 4GL's that go into its pros and cons at great length. Let me just say that many people in the HP applications arena might consider this one application tool the greatest asset they have in their tool box today. Many companies could not function with out their 4GL for new development or more importantly troubleshooting, production fixes and adhoc or information center reporting. Later on I will show how a 4GL can be a major tool in creating some of our "Tinker-Ware" processes for maximum investment.

P.C. Revolution

Remember the HP2622, HP2624, HP2626, well some people may not be familiar with them since are disappearing from many desks. The introduction of the P.C. as we all know has revolutionized the world. How did it affect our world of HP development ? The P.C. revolutionized our HP tool box in countless ways. The first is hardware capabilities. Now we could choose between centralized or local processing depending on what seemed more advantageous. The role of the P.C. and its integration with other tools will be looked out in more detail later on. Up until the P.C., office automation techniques were not the most fun to perform on the HP (be honest). The world of office automation was added to our tool box due to the P.C. Word processing, graphics and financial spreadsheets are now almost as vital to companies as their HP software. The tool that really can link all of these P.C. and mainframe tools together is the terminal emulator packages that have been developed which enables our P.C. to act as an HP terminal. The practical usages of this tool will be shown later on in real examples of "Tinker-Ware" processes that have been built utilizing mainframe and P.C. tools. More and more the concept of the personal computer workstation is becoming the norm rather than the exception around HP shops. The P.C. gives the developer far more greater tools to work with in conjunction with his HP3000 machine while eliminating the need for two machines on a desk.

New Wave

This is the present era. I label this New Wave not specifically for the product itself, rather for what the concept represents and how it has broaden our applications tool box. This era is marked by the ideas of objects, Executive Information Systems, compound documents and multivendor networking of applications. The tools developed through the years have made the ways information can be retrieved infinite. Due to diverse processing needs, companies now find themselves not only with HP classics and spectrums but with important data residing on LAN servers and different vendor mainframes. This data may reside locally on site or at remote locations across the country or the world. The challenge to us as applications developers is how to leverage off the multitude of hardware and software options available to us with our HP systems as hubs to build new processes. The next section will focus on considerations when building an applications tool box for your company.

BUILDING THE TOOL BOX

Companies often buy products without fully realizing the countless ways the product can be utilized. This applies to the capabilities of the product itself as well as how it can be integrated with existing tools. The following suggestions before one buys applications tools and after the purchase has been made may help save your company many hours of time, effort and money.

Take Inventory

Compose an inventory by functional area of what application tools you have in the following arenas:

- Word Processing
- Spread Sheets
- File Utilities
- Text Editors
- Third generation compilers
- Report Writing tools
- Terminal Emulators
- Screen generators
- File transfer protocols
- Database management tools for capacity changes, structural changes etc.
- P.C. data base systems
- Graphics (HP and P.C.)
- Networking (RJE etc.)

These are just some of the major areas to get you started.

Understand Capabilities

Besides knowing what you have understanding the capabilities of these products is really the crucial issue. For example: many companies know they have a tool say Reflection. They know it is a terminal emulator. Yet do they know that it has its own programming language. Maybe the programmers know it but not the system analysts. It seems that from management on down to the junior programming level, everyone knows a piece of the puzzle but not the full picture. By knowing that Reflection has a programming language, the system analyst could create a "Tinker-Ware" process to programmatically transfer a large file to to a P.C. from the HP for incorporation into a graph. By not being familiar with the tool, he might recommend buying an HP graphics package or that a user manually input data into a graphics package.

Share the Wealth

Once a product is purchased, take the time to properly train personnel on the product and its capabilities. This training will foster the sharing of ideas among people. The more people who understand the capabilities of a product, the more potential "Tinker-Ware" applications that will be developed. New "Tinker-Ware" processes are the result of people being creative with the tools they have. They can't be creative if they are not confident of what their tools do and do not do.

When taking inventory and documenting capabilities, do it in teams or even as a whole department if feasible. You will find that once again many people may have knowledge of only pieces of the puzzle. Once the list is complete, distribute it to all personnel involved to act as a reference guide. Like all documentation, keep it current as new applications tools are purchased or new "Tinker-Ware" processes are developed.

There is no generic formula for when and what products you should buy for your tool box. Application needs and company budgets are obviously the driving factors. However, by following some of the suggestions outlined above, the decision will be a more informed one. By taking inventory and understanding the capabilities in the functional areas, (Word Processing etc.) the holes will become apparent. Whether these holes need to be filled by combining existing in house processes to form a "Tinker-Ware" process or to go outside for new software will now become a more thought out decision. The key here is before spending additional dollars, exhaust all of your inhouse options of integrating existing tools. Creative thinking will pay off. Try it, you'll be amazed at the benefits and dollar savings.

Getting started is always the most difficult phase in many projects. The next section will focus in on some "Tinker-Ware" possibilities through integration of different tools. The section on case studies will provide detail of actual processes that have been built as a result of the concept of applications tool box integration.

INTEGRATION

These examples are based on actual experience over the years and once again is not intended as an endorsement of any particular product.

- **Supplementing Applications software packages**

When buying off the shelf applications software (G/L etc.) certain areas never seemed to be satisfied. Often this falls into the area of reporting. Having a report writer in your tool box might alter your thinking in making the purchase. The best way to explain this is through an example. When purchasing a Fixed Asset software package, usually the main goal is to automate the depreciation function utilizing several methods.

Computer Museum

Most packages have similar functions in this area. The difference comes in the report writing capabilities. In many cases it is more cost effective to select the package that can do the basic calculations and supplement the reporting function by developing your own reports. Why pay for a report writer when you can create your own "Tinker-Ware" process to integrate the report writing function with a screen (menu generator) along with the application's database and other file structures. Cost savings could really be appreciable.

o **Automation of files for Auditors**

Here is an idea to score some points with your outside auditors (which everyone can always use). Auditors often require test files of your data so they can run their independent verification. By creating a "Tinker-Ware" process they can receive files on a regular basis with little or no manual intervention. This process was performed using a 4GL (PowerHouse) along with a terminal emulator and its file transfer protocols (Reflection). Quick was used to set up a menu function that when invoked prompted the user for input file names to download. Quick also built a driver file to be used to trigger the download function indicating the actual MPE file names to transfer. Quiz was then used to extract data and format it to the auditors specifications. The results resided in flat files. The user then pressed another key to begin the transfer. Reflection was invoked which read the driver file and kept downloading files until the driver was empty. A simple diskette was then made of the files on the P.C. and given to the auditors. This process shows a very practical "Tinker-Ware" process to satisfy outside reporting requirements. Besides the cost savings in professional fees, the appreciation from the outside auditors in helping them do their job faster was a big plus.

o **Word Processing within Applications & Text Storage in Databases**

In some cases, text is the prime source of data. This data often needs some sort of full screen editing and word processing capability. In addition, this data needs to be stored as part of an IMAGE database. Without buying additional software, a "Tinker-Ware" process was created to dive into our applications tool box and integrate TDP, SUPRTOOL and SPL to create our own in-application word processing on the HP3000. Suprtool was used to extract the data from IMAGE and load it into a temporary file. Through an SPL routine, this file was passed to TDP which opened the file and automatically put the user in screen mode for editing. Once the user was done, the SPL routine closed the file and control was passed back to SUPRTOOL which deleted the original records from IMAGE and reloaded the updated text assigning them line numbers for proper order when being retrieved.

The emphasis on these examples is to briefly show what kind of things have been done through applications tool integration. The case studies will now show step by step in detail how very powerful "Tinker-Ware" processes were developed to meet important user needs without buying specialized software.

CASE STUDIES

Problem: Each month we need to download General Ledger account balance information for a selected dept or division from an IMAGE data base into a Symphony/Lotus spread sheet for analysis. The entire process needs to be controlled and run entirely by the user.

The following are the major steps to accomplish this task. The actual code to perform this is also presented. If you wish to try this at your shop certain things might need to be altered slightly. Even if you do not have one of the tools mentioned, reach into your applications tool box and select a substitute. Remember, "Tinker-Ware" is built on creativity.

Step 1 A Quick menu was developed to make the process menu driven.

Step 2 A QUIZ program was developed to access a financial database prompting the user for which year and general ledger period (month) they wanted chosen. The output was written to a flat ascii file.

Step 3 The same program was then used to display an escape sequence to the screen which wakes up the command language in Reflection on the P.C. The escape sequence sent acts like a udc by executing the commands in a file (glxfer.cmd) on the P.C.

Step 4 The command file programmatically downloads the ascii file created with Quiz and gives it a name in a P.C. directory that Symphony can recognize. Symphony is then invoked with an initialization routine to enable Symphony to read a macro library.

Step 5 The user executes an individual macro which imports the file into the preformatted Symphony worksheet.

Note: You will notice that SPOOK5 is invoked from Quick along with the command file transmitting via 'PTRANSMIT' an "Exit" command back to the HP3000. This logic is necessary in order to accommodate the execution of the protocol program PCLINK.PUB.SYS. This program which transfers the files in Reflection needs to be run at a colon prompt or within a program such as SPOOK that accepts a RUN command. SPOOK5 simulates this condition within Quick.

```
screen glsend menu

title " 01 General Ledger to Lotus/symphony transfer"
;
;
;
;
;
;
;
;
;
procedure designer 01

    begin

    clear screen
    run command file quizuse=qzgl"
    run command "quiz nolist "
    run command "reset quizuse"

    run command "run spook5.pub.sys"

    end
```

```
QZGL:

DISPLAY "EXTRACTING GENERAL LEDGER BALANCES...."
:
:
:
:
:
:
access fin-summary link (year-acct-no[3:24]) to account no of &
  chart-of-accts
choose year parm
set report spacing 0
set page length 0
set nohead
set report limit 10
set report name extfile
set report device disc
define perd num*2=parm prompt "Select GL Period: "
define del char*1='"'
:
:
:
:
report del account-no del del description del summary-amt-1
go
display "Esc&oCCHAIN c:\symph\glxfer.cmd"
exit
```

GLXFER.CMD FILE ON PC.

RECEIVE C:\SYMPH\GLXFER.PRN FROM EXTFILE.PUB ASCII DELETE

CD\SYMPH

SHELL SYMPHONY

PTRANSMIT "EXIT"

General Ledger - P.C. example

These are some of the basic steps that will need to be performed once to automate import of file from within Symphony. The intention here is just to make one aware that the importing of the into the spreadsheet through some basic tasks. Please consult the Symphony/Lotus manuals under MACROS and FILE IMPORTS for more detail and additional steps for invoking MACROS.

Step 1 Attach Macromgr for library initialization
(F9/Application/Attach/MACROMGR.APP/Quit

Step 2 Create Macro (Home/Down 20 etc./Range/Name/Create/Macro name)

Step 3 Load to Library (F9/Macros/Save/'Library name - xxx.mlb)/yes)

Problem: An employee file resides in an IMAGE database. The file contains employee names and addresses. This file needs to be downloaded and incorporated into Word Perfect Mail Merge.

Step 1 A Quick menu was developed to make the process menu driven.

Step 2 A QUIZ program was developed to access the employee database The output was written to a flat ascii file.

Note: One may not be aware that some free file utility software comes with the HP3000. Two such utilities are used in this process. This is a good example of how you can build processes by using free software as part of the process. A program called MAKESD needs to be run once which creates a shell for an SD file (self describing file) which is used in the data conversion. The other will be explained later.

Step 3 The flat ascii file is then copied into the SDfile shell created via the program MAKESD

Step 4 The second free utility is now run, SDTODIF.PUB.SYS. This program converts the file into a DIF format which can be used with the Word Perfect conversion program.

Step 5 Fcopy is then used to display an escape sequence to the screen which wakes up the command language in Reflection on the P.C. like the G/L example. This command file downloads the data file and coverts the DIFile into the necessary Word Perfect format for the mail merge. Last, it calls Word Perfect onto the screen.

Note: Refer to G/L example regarding the use of SPOOK5 in the process.

```
screen wpsend menu

title "SAMPLE MENU" at 2,35
title "01  Create Employee Mail Merge File"
.
.
.
procedure designer 01
begin

 run command &
   "run quiz.current.cognos;auto='qzemp'" &
    clear all nowarn

 run command &
   "run fcopy.pub.sys;info='from=rawfile;to=sdfile; &
    nouserlabels '"&
    clear all nowarn

 run command &
   "run sdtodif.pub.sys;info='sdfile,diffile,1'" &
    clear all nowarn

 run command &
   "run fcopy.pub.sys;info='from=chaincmd;to'" &
    clear all nowarn

 run command &
   "run spook5.pub.sys" &
    clear all nowarn

end
.
.
.
build
```

QZEMP:

```
display "Extracting EMPLOYEE name and address info from DATABASE.."

def comma = index(employee-name,",")
def full-name char*28 = &
  pack(employee-name[comma+1:22] + " " + employee-name[1:comma - 1])
def citystate char*32 = &
  pack(employee-city+" ,"+employee-state+" "+employee-zip)

rep &
  full-name employee-street citystate salutation

set page length 0
set page width 108
set nohead
set rep spacing 0
set rep dev disc name rawfile
go
exit
```

C:\WP50\WPRECV.CMD:

```
receive c:\wp50\diffile from diffile delete
cd \wp50
shell convert diffile empdata a
shell wp
Ptransmit "Exit"
```

CHAINCMD:

```
Esc&oCCHAIN C:\WP50\WPRECV.CMD
```

SUMMARY

Again my children's creativeness in joining together parts from different toys to build new toys shows how we can join application tools to form new 'toys' thus maximizing what we already have. The objectives of the paper were to create an awareness and to spark the creative juices in all of us. By taking time to explore and understand the vast tools that you have, sophisticated tasks can be accomplished which will save many hours of manual effort without necessarily buying additional software. The integration of existing tools to meet user needs is what "Tinker-Ware" is all about and is a concept that can only grow in value as more and more 'toys' become available for us to play with.

Leasing: A Strategy to Reduce Costs and Manage the Future

Paul Perricone
Hewlett-Packard Company
Finance and Remarketing Division
331 East Evelyn Avenue
Mountain View, California 94041

Recently, we heard from a customer who was in a real bind. Two years ago, they needed to expand their computer room. They were interested in acquiring a new technology which was being developed but not yet released. In the interim, they elected to buy an alternate system with the intention of selling it to a broker when the new technology became available. Due to cash constraints, their financial people took out a loan for the equipment.

They are now ready to move to the new technology. Much to their surprise, however, the cost of paying off the loan far exceeds the proceeds they will get from the sale of the existing system. They have the budget to buy the new technology but the acquisition is on hold because they cannot afford to dispose of the existing system!

Where did this customer go wrong? They had done a great job evaluating the cost of operating the computer system. They had looked at price performance, training, maintenance and staffing requirements. They had explored their future needs and factored the cost of the new technology into their budget. How could they have possibly anticipated the disposal cost?

In actuality, this customer could have avoided being painted in a corner if they had done a better job of matching the financing with their long-term system needs. They had excelled at evaluating the impact of the improved technology on their overall system costs but failed to recognize that financial factors can have just as big of an impact.

An effective cost of ownership analysis must cover both the technical and financial sides of the solution. A key success factor in this analysis is choosing a financing method which gives the users the amount of flexibility required to meet their long-term needs at the lowest cost.

This paper will describe how selecting the appropriate financial avenue contributes directly to successful cost management. It will give an overview of the most popular financing options available and how they vary in the amount of flexibility they afford. The focus will be on leasing as a financing source and how the lessee can maximize the cost effectiveness of the system through the careful selection of the lease structure and negotiation of the terms.

Financing Options

To successfully integrate the long-term plans into the financing decision, it is very important to understand the different financing options available. For the purpose of this paper, we will focus on the four most popular ways to finance equipment: (1) cash; (2) installment loans; (3) capital leases; and (4) operating leases.

Cash Payments

Companies which pay cash for their systems are generally either cash rich and/or debt averse. Some people do not consider a cash payment as "financing" as there is no interest charge involved. However, there is an opportunity cost associated with a cash payment. By using funds to purchase a system, there are less funds available to invest in other areas of the operation.

Financial theory suggests that long-term assets, such as computer systems, should always be financed through long-term debt. Cash should be reserved for working capital to fund the daily operations of the company and invest in such areas as R&D. However, if a company has more cash than it needs for operations, it may elect to use working capital to buy long-term assets.

When a company pays cash for a system, it is recorded as an asset and depreciated over its economic life. This accounting treatment is the same for both the financial and tax books.

Installment Loans

Installment loans are designed for companies which want to own the equipment without incurring a large cash payment up front. As it is generally desirable to minimize the monthly cash outflow, a five year term is common on computer equipment. Installment loans are written such that the equipment is completely paid for by the end of the term.

As with a cash payment, the system is recorded as an asset and is depreciated. In addition, the debt is shown on the balance sheet as a liability and the monthly interest charge is expensed. This accounting treatment is the same for both the financial and tax books.

Capital Leases

Capital leases are very similar to installment loans in that the lessee's intent is to own the equipment and they are trying to match the payment with the benefit they receive from the system.

Capital leases are defined by the Financial Accounting Standards Board (FASB) Statement No 13. Basically, FASB 13 states that if the lessee's intention is to own the equipment, they should capitalize it on their balance sheet just as they would for a cash purchase or installment loan. FASB 13 then goes on to give four criteria which distinguish whether a lessee plans to own or just rent the equipment. If a lease meets any of these four criteria, it is considered a capital lease.

Per FASB 13, a lease is to be capitalized if it has either an automatic transfer of title or an end-of-term purchase option below the equipment's expected fair market value at the date that the option becomes exercisable. FASB also considers a transaction to be a capital lease if the lessee maintains possession of the system for more than 75% of its economic life or if the present value of the non-cancellable lease payments exceeds 90% of the system's fair market value.

A capital lease is treated similarly to the installment contract in the financial statement. The asset is capitalized and the corresponding debt is shown as a liability. The value of the debt is calculated using a formula provided by FASB. Each month, the lessee will expense both the interest portion of their lease payment and the depreciation of the system.

One major difference between a capital lease and an installment loan is that a finance lease may have different treatment on the tax books depending upon how it is structured. The IRS has rules addressing the accounting for leases. As FASB and IRS accounting rules differ, it is possible for lessee to capitalize a lease on their financial books and not capitalize it on their tax books.

The gray area between FASB and IRS classifications is beyond the scope of this paper. Suffice it to say that lessees should always consult with a qualified accountant as to both the financial and tax implications of a lease structure before signing a lease agreement.

<u>Operating Leases</u>

Operating leases are geared toward companies which do not necessarily want to own the equipment. The terms of these leases generally span from 12 to 48 months for computer equipment. A monthly rental fee is assessed and at end of the lease, the equipment can be returned with no additional obligation. In most cases, the rental payments will not have covered the full price of the equipment over the life of the lease. The leasing company must, therefore, be able to re-sell the system on the used equipment market to make its profit.

Companies which are concerned about obsolescence or have only a limited need for the equipment usually finance through operating leases. However, there are some companies who want to own the equipment and still use operating leases. They do this for accounting reasons. Depending upon the accounting assumptions they make, they may not be required to record either the system as an asset or the lease obligation as a debt on their financial statements. This "Off Balance Sheet" treatment makes operating leases very attractive to companies who are concerned about the amount of debt they show on their books.

Furthermore, as some companies classify operating lease payments as operating expenses instead of capital expenditures, operating leases also appeal to managers of companies with limited capital budgets.

For the financial books, neither the asset nor the corresponding debt are recorded. Each month, the entire lease payment is expensed. In most cases, an operating lease will receive a similar treatment on the tax books. This is called a true lease - the IRS's equivalent of FASB's operating lease.

However, just as with the finance lease, there are some differences between the FASB and IRS classifications. It can be very costly to assume that FASB and the IRS will always treat a given lease structure the same way. Therefore, consult with a qualified accountant about these issues before signing the lease agreement.

If a company truly wants to own the system, an operating lease is not a cost effective method of financing. The cost of exercising the purchase option makes the overall cost of an operating lease extremely high compared to that of an installment loan or a finance lease. This is the price paid by the lessee for the fact that they took no risk in the future value of the system.

Recent Changes in the Leasing Environment

Both capital and operating leases are gaining popularity as sources of computer funding. A Price Waterhouse/Datamation survey estimated that 35% of all systems sales were leased on operating leases in 1987 and that number would grow by over 11% in 1988. The increased demand for computer leasing can be traced to several environmental changes.

1. Alternative Minimum Tax - The Tax Reform Act of 1986 tightened the provisions for the Alternative Minimum Tax (AMT). The end result of this measure was that corporations could be subject to additional tax liabilities if there were significant differences between the income they reported to their stockholders on their financial books and the income they reported to the IRS on their tax books.

   In most cases, there will be a variance between the profit on tax and financial reports because companies are allowed to use different accounting methods for the two sets of books. Many companies, for example, use a different depreciation method for tax purposes than they do for financial reporting. If the difference between the two sets of books becomes significant, the IRS requires that the income reported on the tax form be adjusted to more closely reflect the income reported to shareholders. For companies on the brink of becoming subject to the AMT, it has become necessary to avoid creating any more of these differences.

   One way to do this is to enter into operating leases. Since the transaction will usually be treated as a rental, it is not recorded as an asset and, therefore, is not depreciated. This off balance sheet treatment averts the depreciation expenses and prevents the potential reporting difference.

2. Computer Technology - A second reason for the growth in leasing has to do with computer technology. The interconnectivity of hardware gives the customer more flexibility in changing hardware vendors. Also, the technology is turning at a much faster rate. Unsure that sufficient upgrade paths are being provided, customers are seeking the ability to walk away from the equipment after a few years. Again, the operating lease is the preferred solution for these customers.

3. Industry Competition - The demand for leasing has also been heightened due to changes within the leasing industry itself. During the early 1980's, the consistent high growth in the leasing market attracted many new players including independent leasing companies, banks, and computer manufacturers. These lessors made huge profits from the tax benefits associated with lease funding.

   With the Tax Reform Act of 1986, many of these tax benefits were repealed. At the same time, the market's growth rate began to slow, reflecting a more mature industry. The result has been a consolidation within the industry and fierce competition for the lessees' business. The Tax Reform Act took away much of the appeal for capital leases for both the lessee and lessor.

   As lessees began to realize the benefits of operating leases, the leasing industry also turned its attention to developing this business.

Looking Beyond Rates

The effect of this competition has created a tremendous opportunity for lessees as operating lease rates have dropped dramatically. At the same time, however, lessees need to look beyond the rate. They need to integrate the user's needs with the financial decision and seek financing structures which will minimize the cost of meeting them.

In many cases, giving the financial decision maker access to the future plans for the system will be a new way of doing business. It will require meaningful communication between the financial and technical decision makers. This may involve a substantial time investment depending on how well they currently understand each other's way of evaluating a decision. The following questions can be used as a basis for opening the communication channel.

1. How long will the configuration being bought today meet the user's needs?

   This question will help gauge the useful life of the system to the user. If the user is viewing it as a long term investment, the financing should be geared toward ownership. However, if the system has a limited useful life, an operating lease may be a better strategy.

2. When the configuration needs to be changed, will it require the return of all or part of the existing system in exchange for a new product?

   This information is necessary in clarifying the issue of useful life. Sometimes a system can be expanded to meet the user's needs by merely adding some additional equipment. For example, the user may tell you that the system can be used for two years before it will need to be changed. At that time, the system will need another disc drive and some additional terminals. It will then meet his needs for another three years. The useful life of the system is really five years and the financing should result in ownership.

   Compare this to the scenario where after two years he plans to replace the system by a new model which will be introduced next year. Given the fact that the current technology is about to be updated and its market value, therefore, driven down, it may not make sense to commit your firm to owning the system. It may be preferable to merely lease the system for the two years.

3. Is there a long-term commitment to the current computer manufacturer?

   This question is becoming increasingly important as the computer industry moves towards interconnectivity. The fact that you can assemble a network of computers from several different manufacturers means that it will be less expensive to switch from one vendor to another.

   There are definite advantages to partnering with a limited number of vendors. A long-term relationship gives the vendor the opportunity to really understand your business and consult with you on how their technology can work to maximize your profits.

   However, occasionally there are other factors in the relationship which are unsatisfactory. There may be problems with service or support. The user may feel uneasy that the current vendor will have the best technology available when he needs it. If he has these types of doubts, he may be thinking of switching vendors.

In this case, it is important not to commit to ownership of the equipment through the wrong financial arrangement.

4. Where is the proposed system in its life cycle?

   This question is extremely relevant if the user is not committed to the current manufacturer. Computer equipment loses market value very rapidly as the technology becomes dated. Therefore, if you are looking at new technology, you may feel comfortable in assuming an ownership position and then selling the system on the used equipment market should the user change vendors. However, if the technology is already mature at the time you obtain it, you may be better off with an operating lease in which the lessor takes the risk on the equipment's future value. The monthly payment will be higher on a short-term operating lease than on an installment contract or capital lease, but the overall cost will be less if the equipment becomes obsolete.

5. Does the manufacturer offer trade-in credits?

   Manufacturers offer trade-in credits as an incentive to purchase new technology. While it is impossible for the manufacturer to commit to a set trade-in credit for an upgrade that will take place three years hence, the technical decision maker will probably have a good idea of the manufacturer's commitment to trade-in credits. This is especially true if there has been an on-going relationship with the vendor. For example, the buyer may know that the manufacturer always offers generous credits when a new technology is introduced as a way to establish market share. The existence of a trade-in credit helps offset the risk of ownership. It ensures the future value of the equipment as you do not need to go to the used equipment market to sell the system. You, in effect, sell it back to the manufacturer.

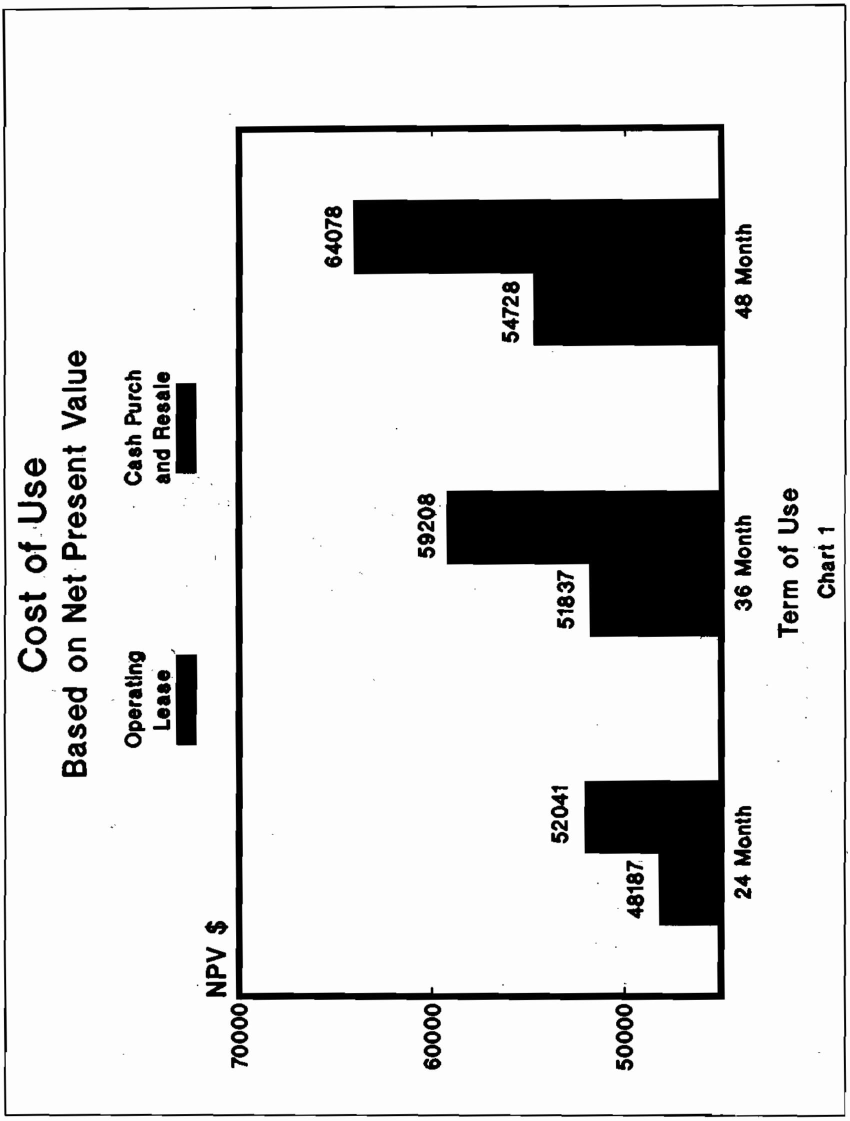
Using Net Present Value Analysis

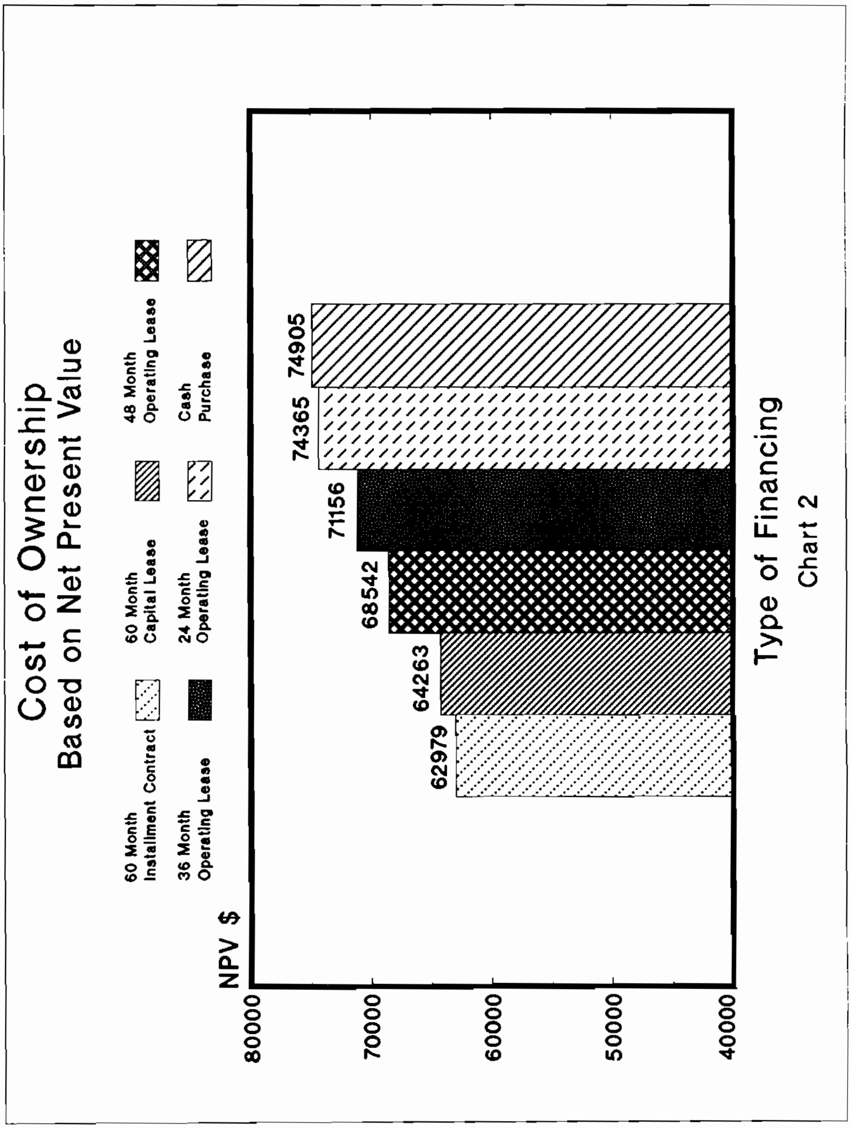
The benefits to be derived from matching the financing to the user's needs can easily be expressed in numbers. To explore this issue, we will be using Net Present Value analysis (NPV). NPV is based upon the premise that a dollar received today has more value than a dollar a year later as today's dollar can be invested to earn interest. Therefore, comparing a dollar received today with a dollar received next year is really comparing apples and oranges. The amount of cash payments and receipts occurring over the course of the year is the annual "cash flow."

NPV analysis takes the cash flows in future years and translates them in terms of today's dollars. By doing this, all of the cash flows have the same base value and can legitimately be compared. NPV analysis is very prevalent in evaluating financing options. The various options result in different cash flows depending upon the timing of the payments and the tax implications.

For example, assume that your firm has decided to acquire a system costing $100,000. The funds have been budgeted for cash payments. You calculate the NPV of buying the system and depreciating it over five years as $74,905 (See Appendix B Case 1).

Now assume that in a conversation with the technical buyer, you discover that they only plan to use the system for three years. By that time, the demands on the system will be such that they can justify investing in a new computer architecture which should be available next year. The NPV of buying the system with cash now drops to $59,208 because


Cost of Use
Based on Net Present Value
Operating Lease
Cash Purch and Resale
NPV $
70000
60000
50000
48187
52041
51837
59208
54728
64078
24 Month
36 Month
48 Month
Term of Use
Chart 1


Cost of Ownership
Based on Net Present Value
60 Month Installment Contract
60 Month Capital Lease
48 Month Operating Lease
36 Month Operating Lease
24 Month Operating Lease
Cash Purchase
NPV $
80000
70000
60000
50000
40000
62979
64263
68542
71156
74365
74905
Type of Financing
Chart 2

you will be able to re-sell it on the used equipment market after three years and recoup some of your investment (See Appendix B Case 2).

The fact that a new computer architecture is imminent, however, should serve as a red flag. Perhaps the re-sale value is overestimated as the new technology may obsolete the existing system. There is now a larger degree of risk in taking an ownership position in the equipment since the user's intent is to dispose of it.

To avoid the risk of ownership, you might consider an operating lease. The NPV of a 36 month operating lease when the equipment is returned is $51,837 (See Appendix B Case 3). This cost is significantly lower than the cost of a purchase and re-sale, and it does not involve making any risky re-sale estimates.

This is not to imply that an operating lease always leads to cost minimization. As previously mentioned, it is not cost-effective to use an operating lease if the user's intent is to own the system. For example, the 36 month operating lease has an NPV of $71, 156 if the purchase option is exercised at the end of the term (See Appendix B Case 4). Compare this to the NPV of a five year capital lease ($64,263) or a five year installment contract ($62,979) (See Appendix B cases 5 and 6).

Chart 1 summarizes the cost of use for a $100,000 system. It compares the cost differential between renting equipment on an operating lease and a cash purchase with a re-sale.

Chart 2 compares the cost of ownership for the same system. In this case, it shows the cost differential between exercising the purchase options on leases with various terms and paying cash.

Lease Negotiations

In these examples, leasing has proved less expensive on an NPV basis than purchasing with cash. In many situations, this will be true. However, in some cases, it will not. For example, if your company has a much lower cost of capital than the implicit rate of the lease, it may be less expensive for you to use cash. Also, if your intent is to own the equipment, short-term leases may be more costly than cash.

Leases may also become more expensive than cash purchases if the lease is not properly negotiated. Just as the financial decision maker needs to ask the appropriate questions of the technical buyer to fit the financing to the system's intended use, he must ask the right questions of the lessor.

It is not good enough to merely shop for the lowest monthly payment. The cost of leasing equipment can be significantly reduced if the lessee negotiates financing terms to meet the user's needs. Again, you should enter the lease negotiations with one of two long-term strategies: equipment use or equipment ownership. You should negotiate the terms focused on this strategy. If the user is not totally sure of whether he will want to own the equipment or return it, choose the strategy which he is leaning towards today but negotiate terms which will allow him to go the other way without substantially increasing the costs.

General Questions:

1.) Is the monthly payment quoted in advance or arrears? The timing of the payments can affect a lease's true cost. Assume that two leasing companies both quote a monthly payment of $2,224 on a five year, $100,000 installment contract. Both leases will start in January 1, but one is in arrears and the other is in advance. The payment for the lease in arrears will be due on January 31. the payment for the lease in advance will be due on January 1. This means that you have access to your money for an additional 30 days with the lease that is in arrears. In terms of Annual Percentage Rate (APR) this translates to rates of 12% for the arrears lease and 12.4% for the advance lease. Therefore, the timing of the monthly payment is a very important factor in comparing lease costs.

2.) Are deposits required with the lease?

Using the above example, assume that the arrears lease requires no deposit while the advance lease requires that the first and last payment be remitted upon signing the lease agreement. What does this do to the APR?

The APR of the arrears lease stays the same. The APR of the advance lease, however, now goes all the way to 12.9%. You should also inquire about the deposit refund should you decide to cancel the order after the lease agreement has been signed.

Questions For Equipment Ownership:

1.) What is the buy-out option at the end of the lease?

Lessors make their profit from operating leases through recovery of the residual value of the equipment. They do this by selling it to the lessee at the end of the lease term or by selling it on the used equipment market. The lower the initial lease rate, the more residual there is to recover. Therefore, if the monthly payment is significantly lower than other quotes you have received, chances are that the lessor is looking to make up for the low rate by charging a higher buyout price.

2.) If the buyout option is priced as "Fair Market Value," how will that figure be derived?

Many lessees feel protected from an unexpectedly high buyout price when they sign a lease agreement with a fair market value purchase option. This purchase option implies that they will be charged the price at which comparable systems are being sold on the wholesale used equipment market. Most lease agreements, however, do not specify who sets the fair market value, how it is determined and what recourse you have if you do not believe that the buyout amount truly reflects the equipment's fair market value. It is important, therefore, to clarify these points before signing any documents. When you and the lessor have reached an agreement on these issues, be sure to incorporate it in writing in the terms of the agreement.

3.) What will it cost if you need to buyout the lease before the end of the term?

As seen on Chart 2, the cost of ownership is reduced by entering into longer term lease agreements. If your technical buyer is sure that he wants to own the equipment but is not sure about when he must have title, you may want to sign up for a long term lease with the ability to buy it out early at no penalty.

For example, the technical buyer has selected a system that can meet his long-term needs through the acquisition of additional boards, Therefore, he wants to own it. However, there is a new computer due out next year which may be even better suited to his needs. When the new technology is introduced, he expects that there will be very attractive trade-in credits. Therefore, he still supports taking an ownership position. However, to get the trade-in credits, he will need title to the leased system. Therefore, there must be a provision for an early buyout.

To be truly safe, the method of determining the early buyout price should be specified in writing in the contract. You should also ask if there will be any additional charges for the early buyout, such as administrative fees, and get these fees in writing.

Questions for Use:

1.) What will it cost to be let out of the contract prior to the end of the lease term?

Most leases are non-cancellable and the lessee is responsible for paying all the scheduled payments. However, on occasion lessees have overestimated the length of time for which they will need a system. It is a good idea, therefore, to find out what will happen if you are two years into a three year contract and the user simply doesn't need the system anymore.

For example, assume that the user has misjudged your firm's growth and that the system is too small to meet his needs. He cannot upgrade it and the value on the used equipment market will not offset the cost of buying the lease out. How much will you have to pay the lessor for them to take back the equipment?

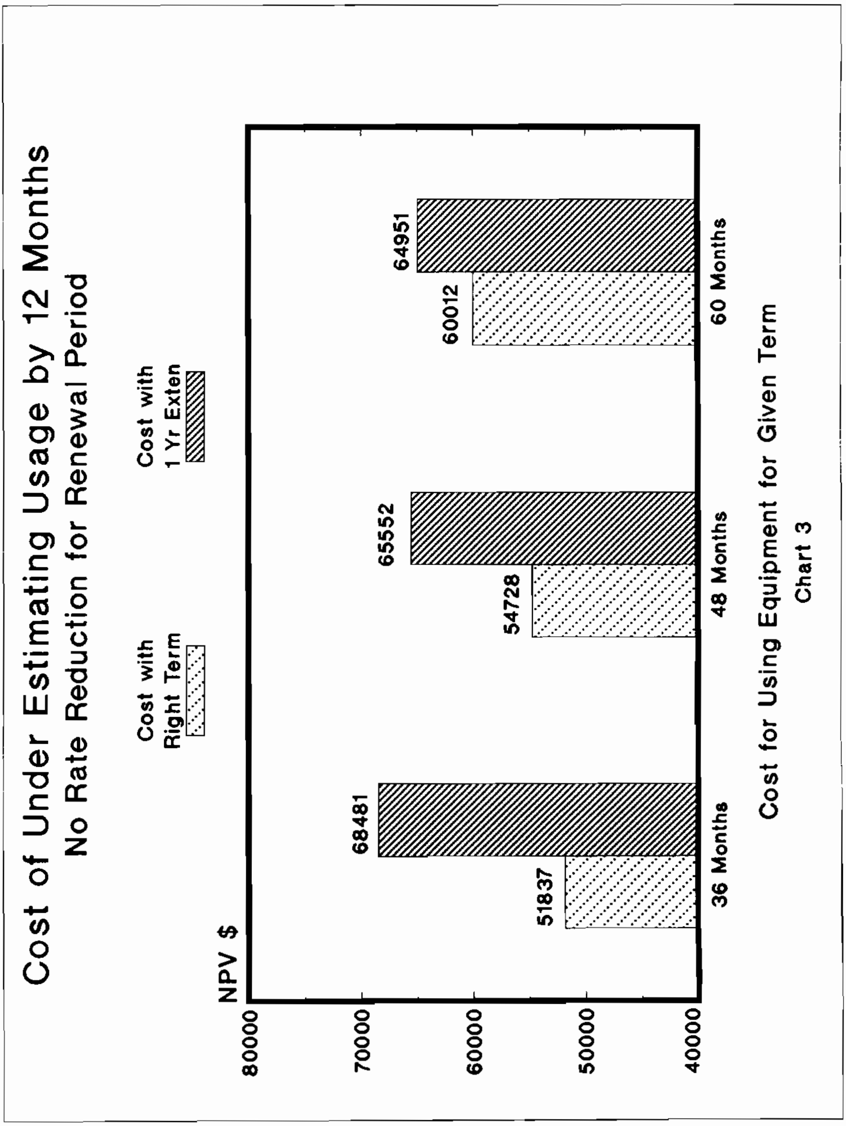
As mentioned, you will be responsible for the remaining payments on the lease, in this case 12 payments. However, if you are going to pay them all at once, you should get some credit as the lessor is going to access to the funds prior to the scheduled payment date. The fact that they can now invest that money and earn interest should compel them to charge you less than the full amount of the remaining monthly payments.

2.) What are the renewal options at the end of term?

Just as a lessee sometimes overestimates the usefulness of a system, they can also underestimate the amount of time for which the system will be needed. The danger in underestimating the lease term is that it can drive up the cost of use dramatically unless the lease is structured with an affordable renewal option.

Chart 1 shows that the lessor recoups more money from the system for longer-term leases. Conversely, the shorter the lease term, the more risk borne by the lessor. Therefore, to offset this risk, lessors charge higher monthly payments for short term leases. If there is no rate-reducing renewal option, the lessee may end up paying the high, shorter-term rate even if they keep the system for a longer term. Chart 3 shows what happens to the cost of use if the equipment is needed for another 12 months and no rate-reducing renewal option is available.

As you can see, if you sign up for a 24 month operating lease and keep the equipment for 36 months, it will cost over $16,000 more than if you had originally signed a 36 month lease. This is almost a 33% cost increase. If a rate reduction had been negotiated into the renewal option, the underestimation of the term requirement would have been much less expensive.


Cost of Under Estimating Usage by 12 Months
No Rate Reduction for Renewal Period
Cost with Right Term
Cost with 1 Yr Exten
NPV $
80000
70000
60000
50000
40000
51837
68481
54728
65552
60012
64951
36 Months
48 Months
60 Months
Cost for Using Equipment for Given Term
Chart 3

Conclusion

An effective system analysis must cover both the technical and financial sides of acquiring a system. More importantly, the two sides must be viewed as interdependent. Successful integration can be accomplished with a three step process: (1) determine the long-term system plans; (2) identify the financing option which will give the appropriate amount of flexibility for the lowest cost; and (3) effectively negotiate the terms of the financing.

The ability of the financial decision maker to successfully complete this process is enhanced by the current lease market. Competitive forces are compelling lessors to help their customers deal with these issues and find creative ways to meet their needs. While leasing may not always be the preferred solution, its potential effect on the overall system cost should always be considered.

APPENDIX A

Assumptions for Net Present Value Analysis

Customer Data:

System Price	$100,000
Cost of Capital	12%
Tax Rate	34% (Total for all taxing entities)
Tax Depr Method	MACRS - 5 Years

Monthly Rate Factors: *

24 Month Operating Lease	3.600%
24 Month Installment Loan	4.731%
36 Month Operating Lease	2.725%
36 Month Installment Loan	3.345%
48 Month Operating Lease	2.275%
48 Month Installment Loan	2.658%
60 Month Finance Lease	2.102%
60 Month Installment Loan	2.250%

End of Term Purchase Options:

Operating Leases	65% Equity Accrual
Finance Lease	10% Purchase Option

Tax Treatment: **

Operating Leases	True Lease (Expense Entire Payment)
Finance Lease	True Lease (Expense Entire Payment)
Installment Contract	Conditional Sale (Expense Interest and and Depreciation)

General Assumptions

1.) All cash flows are calculated on an after-tax basis.
2.) The cash flows are discounted on a yearly basis. The cash flows are discounted in arrears.
3.) The cost of exercising a purchase option is valued net of the present value of the resulting tax benefits. These benefits are gained from the depreciation claimed on the system after the purchase option is exercised.
4.) The re-sale value of equipment bought for cash is based upon a decline in the retail value of the equipment of 20% per year. The equipment is then assumed to be sold at 60% of the current retail price.***

* These rates are for a hypothetical situation. Actual market rates may vary.

** Consult a qualified accountant about the tax treatment of a specific lease structure before signing the agreement. Tax treatment will vary depending upon individual circumstances.

*** The re-sale value assumption for a cash purchase is for demonstration only. This example does not warrant the re-sale value of any Hewlett-Packard computer system.

APPENDIX B

NET PRESENT VALUE ANALYSES

Case 1 - Pay Cash and Keep System

Year	0	1	2	3	4	5
Investment	(100000)					
Tax Effect Of Depr		6800#	10880	6528	3917	5666*
Net Cash Flow	(100000)	6800	10880	6528	3917	5666
PV	(100000)	6071	8673	4647	2489	3215
NPV	(74905)					

Year 1 Depreciation * Tax Rate ($20,000 * 34%)

* Includes the $1,958 discounted for one year. Due to a half-year convention, MACRS 5 year depreciation schedule actually spans 6 calendar years. As this analysis is based upon a 5 year cost of ownership, the tax effect of the depreciation in Year 6 has been discounted back one year and added to the tax effect of the depreciation in Year 5.

PV of $1,958 for one year = $1,749

$1,749 + $3,917 = $5,666

Case 2 - Pay Cash and Sell the System After Three Years

Year	0	1	2	3
Investment	(100000)			
Tax Effect of Depr		6800	10880	6528
Salvage Value				30720#
Taxed Owed on Gain				(653)*
Net Cash Flow	(100000)	6800	10880	36595
PV	(100000)	6071	8673	26048
NPV	(59208)			

Based on $100,000 purchase with a declining retail value of 20% per year. The retail value at the end of three years is $51,200. Assume a broker will pay 60% of the retail price for the equipment (51,200 * 60%)

* There is a capital gain of $1,920 on the sale of the equipment which is taxable.

Sale Price	30720	
Net Book Value	28800	(100000 - 20000 - 32000 - 19200)
Cap Gain	1920	
	*.34	
Tax Owed	653	

Case 3 - 36 Month Operating Lease/System Returned

Year	1	2	3
Annual A/T Payment	(21582)#	(21582)	(21582)
Net Cash Flow	(21582)	(21582)	(21582)
PV	(19270)	(17205)	(15362)
NPV	(51837)		

The after-tax of a year's worth of payments:
(2725 * 12) * (1 - .34)

Case 4 - 36 Month Operating Lease/System Purchased

Year	1	2	3
Annual A/T Payment	(21582)	(21582)	(21582)
Purchase Option A/T			(27142)#
Purchase Option A/T	--------	--------	-------
Net Cash Flow	(21582)	(21582)	(48724)
PV	(19270)	(17205)	(34681)
NPV	(71156)		

The purchase option is based on a 65% equity accrual:

100000 - (2725 * 36 *.65) = 36,235

The system then is capitalized on the balance sheet and is depreciated over the next 5 years. The present value of the depreciation's tax benefit based on a purchase price of $36,235 is $9,093. The net cost, therefore, is:

36,235 - 9,093 = 27,142

Case 5 - 60 Month Capital Lease/System Purchased

Year	0	1	2	3	4	5
Annual A/T Paymt	(16648)	(16648)	(16648)		(16648)	(16648)
Purch Opt A/T						(7491)#
Net Cash Flow	(16648)	(16648)	(16648)		(16648)	(24139)
PV	(14864)	(13272)	(11850)		(10580)	(13697)
NPV	(64263)					

The purchase option is based on a 10% purchase option. The system then is capitalized on the balance sheet and is depreciated over the next 5 years. The present value of the depreciation's tax benefit based on a purchase price of $10,000 is $2,509. The net cost, therefore, is:

$10,000 - 2,509 = 7,491

Case 6 - 60 Month Installment Loan

Year	0	1	2	3	4	5
Annual Paymt	(27000)	(27000)	(27000)		(27000)	(27000)
Tax Effect Of Depr	6800*	10880	6528		3917	5666*
Tax Sav on Int Exp	3959#	3267	2484		1597	592
Net Cash Flow	(16241)	(12853)	(17988)		(21486)	(20742)
PV	(14501)	(10246)	(12804)		(13655)	(11770)
NPV	(62979)					

* See Footnote in Case 1
# The interest portion of the annual payment is expensed creating a tax benefit of:

Annual Interest Expense * 34%

Amortizing a $100,000 loan with a monthly payment of $2,250 and an APR of 12.5%, the first year's interest expense is $11,644.

11,644 * .34 = 3,959

Getting People in Your Company to Use End-user Computing Tools

By Kathy McKittrick
McKittrick Associates, Inc.
5547 South Yampa Street
Aurora, Colorado 80015
(303) 690-1550

There are a wide variety of computing tools available today that are designed to be used by end users in order to increase productivity, make information more readily available and, in general, to make execution of tasks more efficient.

Unfortunately, as so many of us have learned, finding a tool and purchasing it do not necessarily provide us with the promised gains. If users are reluctant to learn and use the product, our return on investment is lost.

This paper deals with the strategies that an MIS department can employ in order to help make users successful in using end-user computing tools.

<u>Selecting the Tool</u> - The first step in selecting an end-user computing tool is to determine what objectives the end-user is trying to meet. This is not necessarily as simple as it sounds. The end-user may say, "I need to be able to generate reports from this system, on my own and on-demand." Your interpretation of this might be "This user wants a report writer. I'll see what's available and decide what to buy. After all, I know where the data is, how it needs to be accessed, and the capacity constraints that we have on our system."

But selecting tools for users without their involvement is like selecting a birthday present for a distant cousin; your chances of choosing something that they want and need are extremely low. And if you don't present them with something that meets their needs, they won't use it.

While users and data processors may may sometimes share understanding of what features are required, they often lack an understanding of additional requirements that each of them have individually; such as, productivity gains, system capacity constraints and aesthetic considerations. For example, the data processing staff is going to be interested in finding a tool that is relatively bug-free, easy to support, and that has minimal impact on system performance. A user, on the other hand, may be primarily interested in finding a tool that can access the sales data base and the accounting data base within the same report, and provide that report within a specific time period.

For all of these reasons, the best approach to selecting end-user computing tools is to apply the same methodology that you would use for system definition and design, except on a smaller scale.

One way to begin is to have the user answer some key questions about their requirements, in writing. Why have them put it in writing? Because it forces them to put more thought into what it is that they want, it encourages them to quantify and prioritize their requirements, it gives you an opportunity to review their requirements objectively, and it can save both of you time. Figure I is a sample questionnaire that may offer you a good starting point.

The presentation of the questionnaire is important, if you are going to get good results. When distributing such a questionnaire, be sure that it gets sent <u>from</u> MIS upper management, <u>to</u> User upper management. A task such as this stands a higher likelihood of being completed if it is presented as a management directive.

The memo which accompanies your qeustionnaire should include an explanation of the purpose, along with a description of the benefits that the user will realize by filling it out. In addition, you should let them know that you will be following up the questionnaire with, what will hopefully be, a brief interview. Figure II shows a sample of such a memo.

When the questionnaires are returned to you, you should look for areas of overlap between users. Do most of the users require multiple data base access? How many of the users have or have used PCs in the past? Are there common complaints about the information that they're currently getting? This will help you to prioritize the features that you'll be looking for in your software search.

There are several things that you'll want to accomplish during the follow-up interview. Obviously, you'll want to get clarification where its needed. You'll want to "play back" your interpretation of the user's requirements to ensure that you are communicating well.

This is also a good time to start educating the user about some of your requirements, such as performance and compatibility. You'll want to do a good job here of translating technical terms so that you don't turn the user off. Never end a discussion of a technical consideration with a phrase like "well, I'll worry about that. You don't need to concern yourself with it." If you do, you may be setting yourself up for later problems when you have to explain why you didn't buy a product that the user favored.

The interview is a good time to prioritize requirements, and to set the users' expectations that any software product will do well to meet 70% to 80% of those requirements.

FIGURE I

The Questionnaire

Name:
Title:
Phone ext:

1. List your areas of direct responsibility:

2. List other areas of the organization that you liaise with most often:

3. List, in order of priority, your business objectives for the coming year:

4. Describe the types of information that would be useful in helping you to meet these objectives, and how often you would require this information (e.g., on demand, daily, weekly, etc.):

5. List information that you are currently getting and ways that this information can be improved:

6. Do you currently have a PC or terminal in your office or home?
   If so, what do you use it for?

FIGURE II

The Cover Letter

To: Margo Johnson
Director of Marketing

From: Bill Porter
Director of MIS

Subject: Management Information Tools

In an effort to better meet your Management Information needs, the MIS department is launching a project to investigate tools that will provide top management with more flexibility and improved turn-around time with respect to information processing requirements.

As a first step in this process, we need to better understand your business requirements so that we can work with you in selecting tools that will best meet those needs. I've attached a short questionnaire which I'd like you to complete and return to me at your earliest convenience. An analyst from my department will contact your secretary shortly thereafter to schedule an brief interview with you. The purpose of this interview is to clarify your requirements and to ensure that we fully understand your priorities.

If you have any questions please contact me at extension 336.

This is also a good time to do an "attitude check". Which users seem anxious for an automated solution? Which users may present road blocks? How can you work with them to avoid road blocks? If and when you find a user who knows what they need, is anxious to pursue an automated solution, and has some on-line experience or native aptitude, you'll have found your "Product Champion".

What is a Product Champion? Its someone from the user community who can help make your choice and implementation of a tool successful. This is a person with whom you will form a partnership that will allow you to communicate effectively with the user audience. As we go through the discussion of choosing and implementing an end-user tool, we'll discuss more about this person's role.

All of the preliminary steps that we've discussed so far (the questionnaire, the interview and choosing a product champion) can help you to avoid a selection process that involves too many people and too many meetings. Such a process can quickly become cumbersome and time consuming, and as a result, decrease enthusiasm.

We won't go into details concerning methods for product evaluation. There are articles and papers available within the industry that have already dealt with that. What we will discuss is the people side of this activity; that is, dealing with the vendors and the users.

When you've analyzed the results of the questionnaire and the interview, come up with a list that outlines your "buying criteria". This list will be used in two ways. First, you can present the buying criteria to your user community so that you have their agreement before you spend lots of time evaluating products. It may be that in your company, upper management will ultimately define the buying criteria. Even so, it helps to involve the users at this point, in order to get them to buy into the project, and upper management can often be influenced by a strong user concensus.

You can also use this buying criteria as a place to start with potential vendors. This will help them to determine whether or not they can meet your needs, and also help you to avoid their enthusiastic explanations of features that you don't care about. I also like to use this list to cut down on my evaluation time. I ask the vendor to highlight the commands or screens within the documentation that meet these requirements. This allows me to focus on those items that are important to me, without having to dig into an index or table of contents that may use terms with which I am unfamiliar. Figure III shows a sample list of buying criteria.

Once you've done some evaluation and narrowed down the field, make arrangements to present several possible

FIGURE III

Sample List of Buying Criteria

- Must allows multiple data base access within one screen or report

- Must offer security based on user, field and value of field

- Must make use of menus and/or function keys whenever possible

- Must be non-command oriented (allowing user to select from lists)

- Must allow for calculations and multiple sort breaks similiar to our current "homegrown" software

- Must offer the ability to limit certain functiona during daytime processing

- Must be within budget

- Must have five references from companies of similar size

- Must have training available

solutions to the end-users. This presentation should include product demonstrations (using data that will be meaningful to the users), a list of pros and cons for each product and a question and answer period. The goal, of course is to walk away with a decision. You may, however, find that, based on their questions, a little more research is required. At the very least, you should set a goal for when the final decision will be made, and agree on outstanding action items.

<u>Product Training</u> - Once you've decided which product to purchase, you should begin to make plans for product training. Product training is a crucial step in your successful implementation of an end-user tool. If users can become proficient fairly quickly (with a minimum of time, pain and embarrassment), then you'll be off to a good start.

With many products, there are several training methods to choose from. You'll need to make your choice depending on the size of your audience, the complexity of the product, and your training budget. In addition, there are a couple of things that you'll want to keep in mind about the nature of your audience.

If you have a large audience to train, then classroom training or self-paced training may be appropriate. If classroom training is presented by the vendor, be sure that the instructor knows who will be in the class, what their level of knowledge is, and how they relate to one another in the work environment. It may be unwise, for example, to team an upper level manager with a subordinate. If the manager is less computer-literate than the subordinate, he or she may be reluctant to fully participate in the class.

Self-paced training can be especially cost effective if you have a large user-community. Moreover, the user is removed from, what may be to some of them, the more threatening environment of a classroom. If you choose self-paced training, be sure that someone is available to help students that "get stuck". This could be a person from the data processing department, or it could be your "Product Champion". The Product Champion will be more familiar with the students' business requirements, and may also be less threatening, due to their natural tendency to avoid technical terms.

One-on-one training is appropriate in some circumstances, particularly when you have a small number of users to train. This can also be an effective method for high-level managers who are reluctant to risk exposing their lack of computer knowledge in a classroom setting.

If you choose classroom training or one-on-one training, be careful is selecting the instructor. If the instructor is an employee of the vendor, get references on the individual instructor; not just the course itself. If it is someone from within your own organization, try to choose someone who has demonstrated their ability to communicate effectively; even if they have less product knowledge than others. A person well-versed in the software can support them from the sidelines.

Product Support - Supporting an end-user tool serves several purposes. Some obvious objectives include answering questions about the product, trouble-shooting and dealing with the vendor. The support of end-user tools, however, needs to be addressed in a more pro-active manner than most support situations, due to the nature of the "customer".

I use the term "customer" on purpose, because when you move into the arena of ongoing support, viewing your users as customers can set a tone that is conducive for mutual success.

Data Processing, in all business entities, is a service organization. In most cases it is considered a "cost center", and as such, is under some amount of pressure to prove its worth. Even after having gone through the steps of selecting the most suitable end-user tools and providing effective training, ongoing support of a product is the crucial piece in getting people to use a product for several reasons.

The first few months of a product's use are critical in its success. I've talked to more companies than I'd care to mention that have purchased products with good intentions, only to let them sit on the shelf and gather dust. A pro-active support team can work to encourage the use of a product, and thereby improve the company's return on investment.

In addition, the user community that you're dealing with is bound to have some personnel turnover. Each time a new user comes on board they'll require some type of training and more hand holding than the more experienced users.

Anyone in the user-support area should view themselves as consultants, who's objective it is to make the user successful. Early on in the use of a product, the support team should not wait to be contacted with problems, but rather seek out the user to determine what they need. Many support organizations consider it to be good news when they don't get any phone calls, but often times, its because no one is using their product.

Support teams can be useful in disseminating information to the user community. If you help one user come up with a creative use of a product, or a work-around in a difficult area, publish your solution in a memo or internal newsletter.

You can also ask users to fill out questionnaires periodically, detailing how they've used the product, what additional capabilities are required, and how support could be improved. This will not only give you the information that you need to do a better job and to lobby the vendor, but it will also give users a legitimate opportunity to vent their frustrations and offer their ideas. Figure IV shows a sample questionnaire.

Quality assurance testing of new product releases can save the support organization lots of time and anguish. When a product is released to the user community with a serious bug, they not only lose time, but you lose credibility. Although vendors have primary responsibility for quality assurance testing, they may not be able to uncover bugs that occur under unusual circumstances that may be present in your environment.

When you release a new software version to your users, it may be beneficial to provide a list of product enhancements that will be of particular interest to them. Vendor documentation that accompanies new releases is often verbose and technical in nature.

We've talked about some of the things that you can do to create an effective support team, but we haven't talked about who the members of that team will be, and here's another area where your Product Champion can be extremely useful. By making the Product Champion the "Product Expert" in the user community, you can help to weed out non-productive or repetitive support calls. Make your Product Champion the first line of support, where all user bug reports and inquiries are funnelled through that Product Champion. Based on my experience, you'll find that anywhere from 30% to 60% of the support calls never get to you, but can instead be answered or resolved by the Product Champion.

<u>Dealing with "Special" Users</u> - I chose the term "Special Users" very carefully, because I feel that we in the data processing business have a tendency to treat a certain type of user with what must be called, at its worst, contempt and at its best, tolerance. A "special" user is one that has difficulty with the use of a product, whether it is due to their lack of experience, lack of enthusiasm, or "techno-fear".

It is important not to ignore these users or simply hope that they will go away. They usually won't. Moreover, they will sometimes lobby others to share their views in order to justify their position. There are several things that you can do to make their lives (not to mention your life) easier.

First, be sure that you offer them one-on-one tutorial sessions with the product. Do this in a private environment where they cannot be disturbed or intimidated by their co-

FIGURE IV

Support Questionnaire

We are providing you with this questionnaire in order to give you an opportunity to evaluate the benefits of *product-name*, and to provide input as to how support services could be improved. Please fill this out at your earliest convenience and return it to *support-person*.

1. I have been using *product-name* for ________ months/years.

2. I have found the most significant benefit to be
____.

3. I have found the most significant difficulty in working with the product to be
____.

4. I would use this product more if ____.

5. Rate the following areas, where 1 is the lowest rating and 5 is the highest rating:

Performance ______

Documentation ______

Functionality ______

Ease of use ______

Support ______

6. Additional comments
____
____
____
____.

workers. Choose a tutor with whom they have a natural rapport, and if that person doesn't work out, try someone else.

The person tutoring the "special user" should be start with very simple features, thereby giving the user an early opportunity for success. Make sure they have one concept down cold before moving on to another. When they get something right, offer praise.

Be willing to show the "special user" your own vulnerability. Offer examples of your own initial confusion and commiserate with them when they don't understand.

Try to focus on product uses that will help them to get their job done more efficiently or more effectively. They'll buy in quicker when they see the benefits.

Many of these suggestions may sound trite and intuitively obvious, but I've seen them work many times. More years ago than I'd like to admit, I gave a software class where one of the students had been a Production Manager for a company for many years. When they bought their computer system, they required fewer Production Managers so they told him he would be their System Manager. Never have I seen a person suffering from more fear of professional failure. He was literally afraid to touch the keyboard, and he was angry and argumentative.

We had one of our support people sit in the class with him as a tutor. I watched her work with this user throughout the week, using some of the techniques I've discussed (and a lot of her own human understanding) and he slowly began to relax. I wouldn't say he had fallen in love with computers by the end of the week, but he had realized a level of self-confidence that he could take with him to his next course and beyond. Today, this man is the Data Processing Manager with responsibility for five Spectrum class machines and over 250 users.

In summary, I'd like to suggest that in selecting end-user computing tools you can maximize your chances of success by organizing the effort as you would a development project. For training and ongoing support, assign specific people to take responsibility for products and their effective use, and measure these people based (at least partially) on user satisfaction with their products. Take a pro-active role in end-user computing and your company will realize many benefits.

Copyright (c) 1990 J.R.Leverentz
James R. Leverentz
President
OFF-SITE, Inc.

INFORMATION MANAGEMENT AND COMPETITIVE EDGE

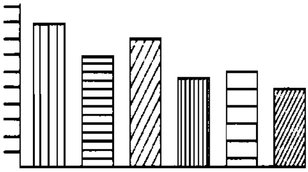
Like it or not, the reliance on electronic data processing is a fact of life. It is no longer acceptable to provide only product to your customers and clients. The next decade will be one of value added services, instant response to customer demand, and exceptional support after the sale. In the next few years, you will have to except this in order to remain competitive regardless of your industry. Once your competition starts using information management to increase their market share, you will have to follow suit. Consider where the competitions increased share will come from.

The general sales manager for a distributor of after market auto parts sums up my point very simply. We sell auto parts, our customers buy same day delivery. They can buy auto parts anywhere. This value added service is made possible in part by computerized inventory control, computer generated delivery routes, and instant order processing. His company maintains a competitive edge through information management.

The demand for improved information management can also come directly from your customers. They are seeking to improve their own response to customer needs and reduce their cost of doing business. Consider the growing move to on time inventory. Instant demand, instant supply. When your customer tells you they will be moving to computer generated purchasing, you had better be prepared to move with them. When the customer is someone as large as General Motors, their attitude will be; If you cannot do this, we will find someone who can! If that customer represents sixty percent of your gross revenue, your inability to manage information has put you out of business.

Just as important as keeping pace with information technology is keeping that technology in place. When a computer generated purchase is electronically issued, you had better be sure your computer is ready to receive. If it is not, the customer will move on to the next available source. This is nothing new. When you need milk and store A is closed you go to store B without even thinking about it. If you are not open for business, the consumer will find someone that is.

Protecting your ability to manage information is called Contingency Planning. Implementing a contingency plan is called Disaster Recovery. These are two different processes. The second is controlled by the first. Contingency planning is the logical analysis and documentation of how you use information. Disaster Recovery is the mechanical method used to protect the critical functions illustrated by the Contingency Plan.

RAW DATA	**INFORMATION**
85940123857439201345872488591029843174238 84930198764830029481882374651423848023999 54537261847326549123766571010938745838411 99483771662545482891008912345342524345163 77487326715612323489072138631456571823521 31241254123654651239872347872134765871623 45123760986776512398732456782824156867243	

For the purpose of contingency planning we must define terms and objectives. We start with Data Processing. In the examples used so far, we have only talked about electronic data processing. Contingency planning must incorporate a more generic approach.

Data processing is the conversion of data into information. Raw data alone is usually meaningless. It must be interpreted. Interpretation does not have to be done electronically. A draftsman using a ruler and pencil performs the same conversion of data as the most sophisticated Computer Aided Design system. The number one rule of computerization applies; if you cannot do it manually, you cannot do it electronically.

Every case of data conversion performed by computers requires manual operations. Contingency planning is for all data processing functions, regardless of the methods used. These non-electronic steps are referred to as the Mechanics Of Operations. In many cases they are more critical to data processing than any electronic process. The reason is, point of contact. Point of contact means the connection between what the computer will use and produce, and our reason for needing it.

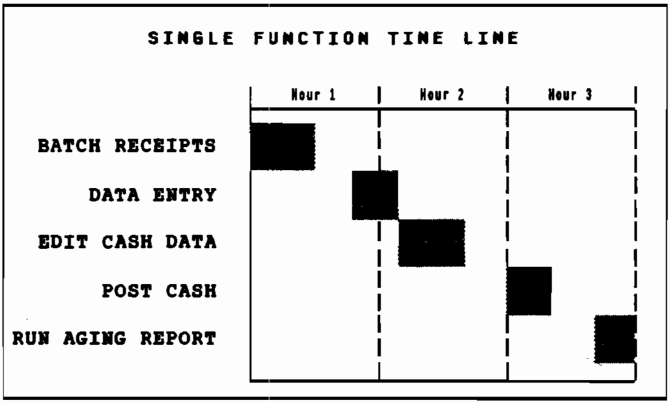
These manual steps include the signing of pay checks, review of in coming invoices for accuracy and customer relations. Even though computers can manipulate incredible amounts of data, they can rarely determine if what they have done is valid. The data provided to the computer must be validated. The results of the electronic conversion of data to information must be scrutinized. Thus, we have the interaction between computer speed and human intuition.

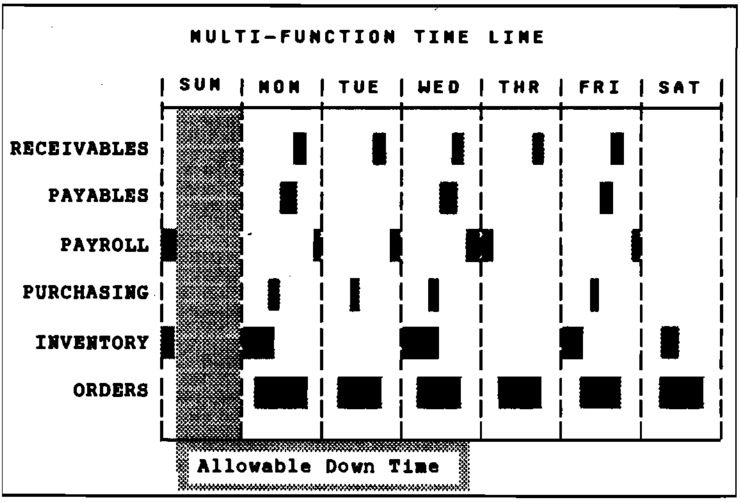
INFORMATION MANAGEMENT		
DAILY FUNCTION	DATA MANAGEMENT	BUSINESS REASON
Collections Purchasing Scheduling	Receivables Supply Demand	Cash Flow Inventory Customers

How much information do you analyze in a single day? As we collect more and more data, we end up sifting through endless streams of computer print out. If we are to prepare a contingency plan to keep the stream flowing, it is necessary to examine the information and determine if it is truly important.

Data processing is a function of availability and time. When you need information, you must determine when the raw data will be available; then, how long it will take to prepare, validate and convert to information. Once the conversion to information is complete, you must decide if the results are valid and critical to the business.

This process of availability, conversion and interpretation is repeated each time new information is needed to support decision making processes. As more and more critical conversions are performed on a regular basis, a schedule of data processing evolves. This time table of critical processing is vital to contingency planning. It documents the critical preparation of information.





If you graphically illustrate the critical processing into a time table you will find gaps of time between critical functions. These unused time spans represent allowable down time. Total allowable down time is defined as the longest period between any two critical processes. This gap of idle time plays an important role in the ability to stay on schedule. Should an interruption in the critical processing schedule occur, the allowable down time is available to catch up on lost work.

Allowable down time must be considered a limited resource. As more critical functions are added to the schedule, the allowable down time is decreased. It is common for critical functions to overlap. This must be taken into consideration when determining the tolerable amount of down time. This determination quickly illustrate just how limited a resource allowable down time is.

If you have one thing to accomplish in one minute and cannot get it done, the next minute you will have two things to finish and one minute to do them. Eventually, you reach a point where it does not matter if anything ever gets done.

When an interruption lasts longer than the allowable down time, something else must happen. The inability to rectify an interruption in critical processing, prior to using up the allowable down time, results in a disaster situation.

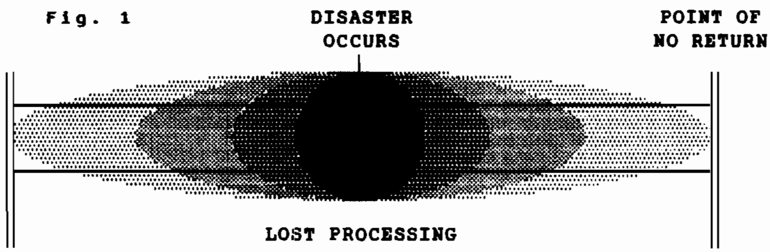
The word disaster indicates destruction, acts of God. You could spend a week making a list of events which can disrupt data processing. For the purpose of Contingency Planning a disaster is simply the inability to stay on schedule regardless of the cause. It is not inconceivable that the event causing the interruption appears to render the allowable down time meaningless. The actual destruction of the computer equipment by fire leaves us with nothing to repair. We do however, still have the period of allowable down time available for determining what we can do once the Disaster occurs.

The inability to react in a disaster situation results in a catastrophic event. Without a Contingency Plan, the process of determining actual losses and restarting the critical processing cycle will be very time consuming. Since data processing is a function of time, each lost minute is amplified by the next. I have one thing to do and one minute to do it.

To illustrate the distinction between allowable down time, disaster situations, and catastrophic events; consider the following three situations.

SITUATION 1: The ABC Company has ignored all the basics of information management. They do not perform back-ups of data and consider the chances of a computer failure to be so remote, that there will not be any impact on the business. They have not prepared any type of standards for using the computer system even though their independent auditors have made recommendations to do so for the past three years. The management of the company considers the computer system to be equivalent to a home appliance, and in private actually wish it were not being used.

Fig. 1



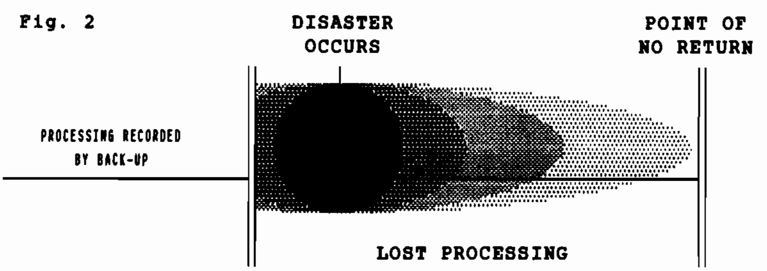
Since data processing is a function of time, disaster situations can be represented as waves similar to those made when a stone is dropped into still water. If a computer failure is represented by the stone, the effect can be seen to expand. The expansion of this failure wave moves both forward and backward in time. This shows that not only has the ability to continue processing data been lost, but everything done in the past is also lost.

To put this into a proper perspective, consider what must happen to recover from the failure. Rework! It is now necessary to pick some point in time prior to the failure, and redo everything required to restore the computer to the current position in time. That current position is always moving forward. The result is financial losses through redundant labor and lack of computerized benefits.

Depending upon how computer dependent the business is, the time scale can be hours, days or weeks. The more dependent, the smaller the scale. Just as with dropping a stone in water, the type and scope of disaster determines the size and force of the failure wave. With no back up dam in place, the failure wave moves freely in both forward and backward directions. If the stone represents a computer virus which has erased all data from the system, the backward effect of the wave reaches to the first time the computer was used. The degree of dependency the business had on the lost data determines the failure waves impact on the future of the company.

SITUATION 2: One of the main points made in the management letter to the ABC Company was data back-up standards. Since no new equipment would be required, the cost to implement regular back-up procedures was minimal. Management agreed that full back-ups of the computer system would be done regularly.

Fig. 2



Using the same failure wave example, what happens when a failure occurs. The point in time of the last back-up becomes a break wall. It is a definite point in history where the need to perform rework ends. The recovery cost savings are obvious in comparison to the cost incurred in situation one where back-ups were not done. A quick look will tell us exactly what processing has been lost. We now know what needs to be done to catch up to the current point in time. That point in time is still moving.

Good system back-up procedures provide a barrier in time. The backward effect of the failure wave is stopped at the point in time of the last available back-up. The more frequently back-ups are performed in the critical processing schedule, the less time required to recover lost data and processing.

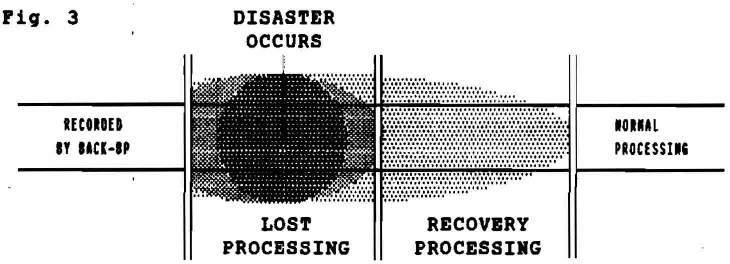
The frequency of back-ups can be theoretical determined by matching the time required to recover, with the allowable down time. If eight hours of allowable down time exists per processing cycle, a back-up should be done every eight hours. In theory that is fine. The real world is a different matter.

Let us say that ABC Company works a five day week. That leaves two days per weekly cycle of allowable down time. If they perform back ups at the end of each work day, then in theory they will have the weekend to make up lost processing. This is true if the failure occurs on Friday morning. By Monday they should be current providing that the disaster has an impact of less than twenty-four hours. If a disaster occurs on Monday, just before back-up, with an impact of seventy-two hours the recovery time is compounded by the forward progress of the failure wave. Not only is there a loss of past processing, but current and future processing are also lost.

SITUATION 3: ABC Company was also told that a contingency plan of action and documented disaster recovery procedures should be developed. Since we can build a break wall in history why not put one in the future.

A back-up of history is a fixed thing. The future presents a problem. If we have not done the processing yet, we do not know where to put the break wall. Unlike the back-up break wall which stops the counter clockwise failure wave, the best that can be expected of a contingency plan break wall is to slow down the forward progress of the failure wave.

Fig. 3



ABC Company decided to prepare a complete contingency plan. They began by identifying exactly what the computer system was being used for. Management was a little surprised at the findings. Their attitude of ..hands off.., changed when they discovered the financial benefits derived from using the computer. Once the contingency planning analysis was complete, they prepared the required disaster recovery procedures. The procedures to keep critical processing on schedule regardless of what caused the failure wave.

It is impossible to stop the wave at a predetermined point in the future, but having the recovery procedures ready and waiting, it is possible to reduce the impact of the failure wave and avoid the point of no return.

Contingency planning and disaster recovery procedures act as break walls in the critical processing schedule. They serve to slow down the forward motion of the failure wave, thus reducing the recovery time required to become current.

When ABC Company was hit by lighting on Monday, just before back-up, there was extensive damage to the computer equipment. The recovery procedures called for the use of alternate computer hardware. The contingency plan pointed out exactly what processing had been lost and recovery started as soon as the alternate hardware was available. The plan also showed what critical processing was not dependent on past processing. Portions of the current processing began right away. As more of the recovery was completed, current processing was added. It was possible to become current prior to the point of no return.

The concept of protecting the critical processing schedule seems to be intangible. We must look at the reasons for having a schedule in the first place. What are the tangible results of the processing schedule and what are the tangible affects of the failure wave?

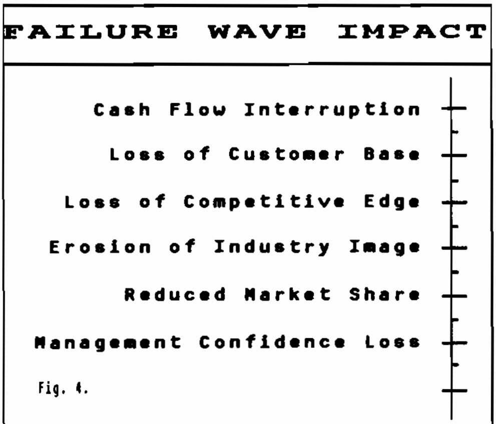


Fig. 4.

Of 150 responses to a survey, all listed the items directly impacted by down time, in the same order. The variable is the span of time between items, which changes according to the degree of dependency in any one area. It is probable that more than one area will be impacted at the same time. By relating your business to this list, it is possible to determine your point of no return in the time line.

In 1987 Texas University conducted a survey of companies which suffered long term computer down time. They were asked to list the affects of the computer down time in order of occurrence. Fig. 4 presents the results. Note that all of the items listed are related to finance and company image. No one indicated that the computer hardware was a loss. One respondent added that after five days of down time the companies prognosis for survival was 20 percent.

To illustrate the failure wave in real terms, consider the first item in Fig. 4. Cash Flow. In the event that you are unable to invoice customers for a three day period, and you are lucky enough to have customers that pay every thirty days, the failure wave extends thirty-three days into the future.

The impact list is useful in determining where your point of no return falls on the time line. For example, let us say that ABC Company is in a highly competitive business. Loss of customer base may be the final breaking point. Once customers have switched suppliers the remaining known items become academic.

If ABC Company is in the insurance business, or some other regulated industry, legal and regulatory violations could mean the end. The frequency with which these rules are monitored defines the point of no return.

What if ABC Company were a law firm. After the disaster which destroyed the ability to prepare for court, it would be necessary to file for a continuance on all cases. Erosion of industry image could be the end.

With the exception of cash flow interruption, and some special cases, it must be noted that none of the reported losses are insurable by any commercial policy. Regardless of the cause, the impact on the business will always be the same. It is that fact which makes contingency planning and disaster recovery possible and necessary.

The objective of contingency planning is the prevention of a catastrophic event. To prevent something from happening, you must know when it will occur. If our contingency plan has determined that allowable down time equals six hours, then when an interruption occurs we know that a disaster situation is six hours away. In other words, we have six hours to prevent the Disaster.

The first consideration is whether or not the down time can be rectified. This is the only time in the recovery process where the actual cause is considered. If the cause is an unexpected power outage, minor hardware failure, or environment control failure, you may be able to estimate or determine what the repair time will be. If the down time is caused by a more serious event such as fire or PCB contamination, the allowable down time has not been lost. We know we have six hours of reaction time.

Regardless of the cause, we can now calculate very closely the point in time when the catastrophic event will occur. If implementation of our Disaster Recovery Procedures requires twenty-four hours, a catastrophic event will occur in thirty hours from the start of the allowable down time. Since the Contingency Plan has documented what is critical, who is critical, and when they are critical, we now have what we need to prevent the pending catastrophe. The contingency plan has provided us with what needs to be done and how much time we have to do it.

The critical processing schedule is what we are protecting through Contingency Planning and Recovery Planning, yet there are two main reasons for business losses caused by computer down time. First is the absence of a contingency plan, Second is an outdated contingency plan. Both reasons are the result of the conventional approach the disaster recovery planning.

CONVENTIONIAL CONTINGENCY PLANNING

Until recently, Contingency Planning has been considered a computer hardware problem. This attitude in the board room causes two distinct problems. First, it places contingency planning in the hands of computer technicians instead of business managers and second, reinforces the anxiety of the work force by placing more importance on computer hardware than people.

It is difficult enough to convince workers in new computer installations that the computer will not replace them, but, make them more productive. Why then, do we insist on just the opposite through our actions. Tom Peters, in his book, Thriving on Chaos, Harper & Row, states, Despite lip service about people, as our most important asset, we value hardware over people, and have done so for the last century. His reference is to employee training, but applies equally to conventional contingency planning.

This Hardware vs. People conflict and misguided assignment of responsibility, comes from a lack of understanding on the part of the Corporate Manager for the true benefits of electronic data processing. The misunderstanding is not entirely managements fault. They may suffer with a false sense of security regarding Business Interruption Insurance that is supposed to protect the company from lost income.

Unless specifically written, interruption insurance is protection against the inability to service customers, not incompetence. For example, if you are unable to deliver goods to your customers, your interruption insurance will provide cash to farm out the work. This will keep your customers satisfied.

What if you also lost the ability to process customer orders? How will your insurance protect you from not knowing what work to farm out? I spoke earlier of a manufacturer who lost sixty percent of his business because he could not take an order. What happens when the customer does not return?

The combination of Hardware vs. People, lack of committed understanding, misplaced responsibility, and false security results in little or no contingency planning. This alone reinforces the need for contingency planning.

"Disaster recovery?

The computer room is down the hall."

Solving the problems of conventional planning starts with realigning responsibility. You must make it known that every function of the business is just as important as the next. If this were not true, given the high cost of doing business, you probably would not perform the function. Responsibility can be equally divided into three groups.

Management Commitment, Human Resources, and lastly Data Processing. If you had to place a priority on any one group, it would be Management Commitment. Contingency Planning is no different than any other business planning project. It will be as good as the effort made.

MANAGEMENT

HUMAN RESOURCES

DATA PROCESSING

For management to commit to a project it must be presented in terms they understand. The most common term in business is Money. In dollars and cents, what exactly is contingency planning all about?

Management commitment to the contingency planning process must begin with a financial evaluation of the current and forecasted situation. For management to view the planning process as a priority item, it is necessary to approach the problem in terms they understand. Since the primary function of management is controlling dollars, the need for a contingency plan must be placed into those terms.

We start with a review of basic accounting.

Is data an asset? If so how much is it worth?

HARDWARE - POTENTIAL

When computer equipment is purchased, it is considered a capital investment. Like all other assets it becomes part of the balance sheet and is depreciated over its expected life span. Assets are items which provide strength to the company through their usefulness. Yet, computer equipment alone provides no benefit. It gives us only the potential to process large volumes of data. Computers are incapable of anything unless a set of specific instructions are added.

SOFTWARE + HARDWARE = ABILITY

These instructions are called software applications. Complete software systems can range is price from $9.95 for household budgeting to hundreds of thousands of dollars for manufacturing process control. Regardless of the application, signed licensing agreements for software show as an asset on the balance sheet. The combined elements of hardware and software now provide us with the ability to process large volumes of data. This ability in itself does not produce tangible results.

The equation, hardware plus software results in ability. Ability as a rule is not an asset. Another factor must be added to the equation. That factor is data.

HARDWARE + SOFTWARE + DATA = RESULTS

The cost of introducing data to the Information Equation is expensed through the wages paid for the service. Data entry costs are far less than the value of the information being gathered. This is even more evident when data is gathered from multiple sources and combined for decision making.

Based on the information equation, we can no longer say that computer hardware is just a dumb machine. When software is added the computer becomes an educated machine with no purpose. In order for the computer system to be a true asset, data must be introduced to the equation. Therefore, data must be considered an asset. As every accountant knows; you cannot add an expense to the balance sheet.

A good illustration of this is in manufacturing where computer systems are used to maintain specifications for raw materials and production process steps. If a manufacturer produces one hundred individual finished products, and, each product is made up of seven different component parts, it is necessary to keep track of eight hundred different items.

In making each of the one hundred products there are an average of twenty-four processes. Each process involves two steps. This results in 4,800 instructions. The total volume of related data is 5,600 entries.

Similar examples of complex data structures exist in every industry. The value of the data can be expressed as at least the cost of manually preparing it every time it is needed. Think of the man hour cost of data reproduction in the middle ages as performed by monks. Now compare it to the electronic methods used today.

If the combination of computer hardware and software are viewed as a production machine, then data must be considered a raw material. Any inventory of raw material has value. The real problem has been in determining just how much value.

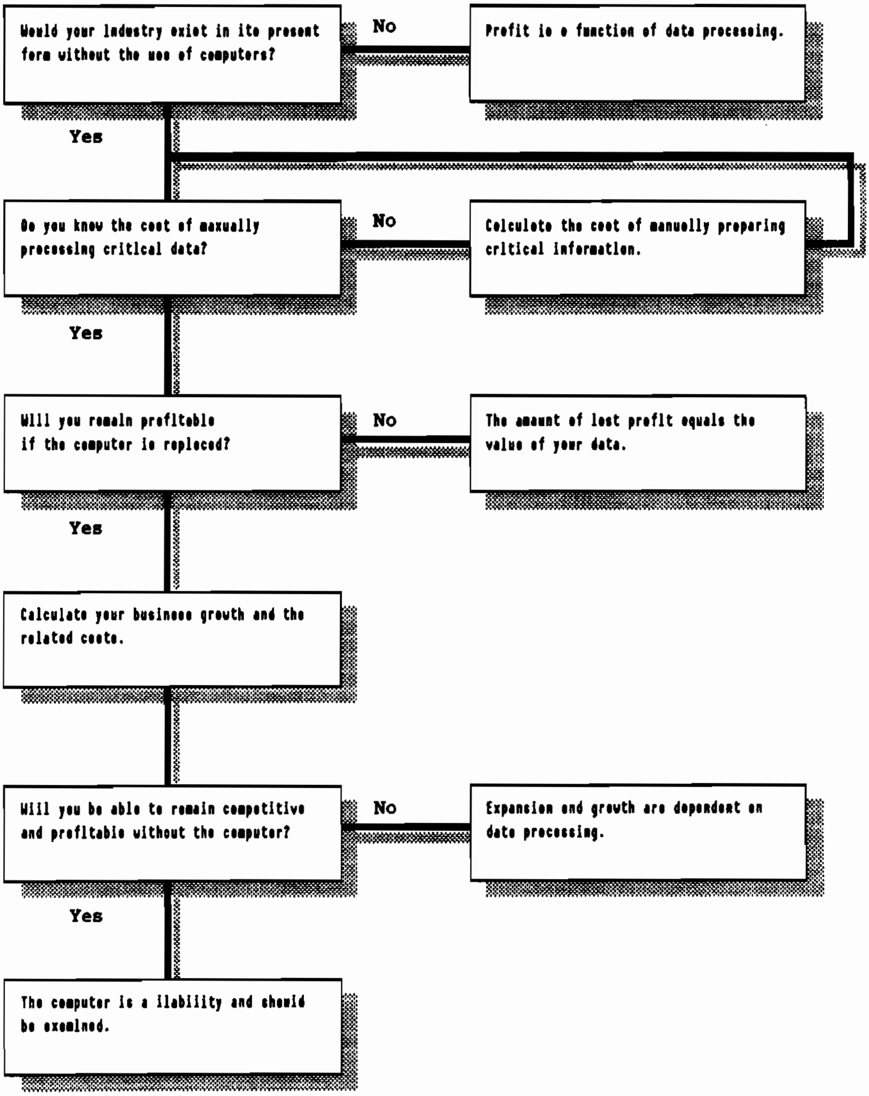
CALCULATING THE VALUE OF ELECTRONIC DATA PROCESSING

Placing a dollar value on information assets is a difficult and sometimes vague task. It begins with relating data processing to profitability. The following exercise is designed to aid in this task. It is based on the principle declaring that, since information management is so expensive, why do it.

The exercise involves answering four management questions. It will be necessary to collect basic financial information about your company. Worksheets with examples are provided to aide in the calculations.

The results of the exercise are useful not only for determining the need for contingency planning. They can be used as a guide for improving computerized productivity and in the cost justification process for new or additional computer system and resources.

VALUING DATA PROCESSING BASED ON CONTRIBUTION TO PROFIT



5192-14

Would your industry exist in its present form without the ability to manage information electronically?

This does not mean that if you are in the computer business your answer is no. You must determine if your business is a result of your computerized information management. This may apply only to a portion of the business. If so, the profitability of that portion can be said to depend on data processing.

As an example, consider the Electronic Malls available from commercial computer access services. Until the advent of the home computer they did not exist.

If you find that your business results from information management, then it can be said that your data is equal in value to your gross profit. Should you determine that the industry would exist regardless of computer usage proceed to question #2.

What would it cost to replace critical data processing with manual procedures?

When we discuss human resources, we will see the specific relationship between personnel and computers. To answer question #2 we must look at personnel from a broad perspective. If you attempt to reduce dependency on computer based systems, that dependency must be placed some where else. Human Beings are the only superior alternative that can accept this responsibility.

In order to calculate the cost of additional payroll, floor space and other expenses required to replace a critical computerized functions, we can use the law of averages. The data required to answer question #2 is readily available from the payroll and accounting departments.

Average Annual Salary	Salary Benefit Expense	Average Annual Wage	Wage Benefit Expense	Payroll Tax Percentage	Overhead Per 81 SqFt.
$27,000	$3,000	$18,200	$1,500	11.42%	$576

APPLICATION	S	W	PAYROLL	BENEFIT	TAXES	SPACE	TOTAL
Accounts Receivable	2	2	90,400	9,000	10,324	2,304	112,028
Accounts Payable	1	2	63,400	6,000	7,240	1,728	78,368
Payroll	2	2	90,400	9,000	10,324	2,304	112,028
General Ledger	3	1	99,200	10,500	11,329	2,304	123,333
Order Processing	2	5	145,000	13,500	16,559	4,032	179,091
Scheduling	2	2	90,400	9,000	10,324	2,304	112,028
Inventory Control	1	3	81,600	7,500	9,319	2,304	100,723
Purchasing	2	1	72,200	7,500	8,245	1,728	89,673
Customer Service	1	3	81,600	7,500	9,319	2,304	100,723
Total	16	21	814,200	79,500	92,983	21,312	1,007,995

Prepare a list of the computerized business functions which are critical. You must exclude any non-essential processing which is done for convenience. In almost every company there are reports of identical information presented in multiple formats. Without the computer you would not take the time to do this. Determine which version is the most useful. Remember the limited resource, allowable down time. Answering question #2 may help you increase computer availability without investing anything but time. That is a return on investment.

It is likely that some computerized functions can not be done manually. In that case, based on how critical they are, more consideration must be given to question #1. Large scale functions such as material requirement planning (MRP), and stock market performance analysis, may need to be broken into smaller fractions. These fractions are analyzed for acceptable levels of manual preparation.

Careful consideration must be given to the numbers of salary and hourly employees required to perform critical data processing manually. If you currently use the computer to print four hundred invoices per day and each invoice includes 40 words, a typist at forty words per minute will require seven and a half hours to perform the same task. Computers operate at high speeds and are designed to organize large volumes of data with endless repetition. Human Beings grow weary of repetitive work and efficiency decreases.

Will the business continue to be profitable without the use of electronic data processing?

The accounting department will have to supply some more data to answer this question.

Last Year Gross Revenue

Last Year Data Processing Expense

Last Year Operating Expense

If you determine that it is not profitable to reduce computer usage, then the amount of reduction in profit can be considered the bottom line contribution of data processing to profit. The computer has now been shown to be a profit center.

If the adjusted gross profit is still acceptable proceed to question #4.

Gross Revenue	38,500,000
Operating Expense	-26,750,500
Data Processing Expense	-550,000
Actual Gross Profit	11,199,500
Add Computer Expense	+550,000
Subtract Manual Expense	-1,007,995
Adjusted Gross Profit	10,741,505
Computer Contribution	457,995

Will the business remain competitive and continue to grow if the computer system is eliminated?

You must now account for the expected growth of the business. This requires the following information; Revenue Growth This Year & Next Year, Inflation This Year & Next Year.

It must be recognized that the majority of data processing expenses are fixed and will not substantially increase with a marginal increase in demand. Manual operating expenses on the other hand will certainly increase proportionally with expanding business.

The following assumptions are made:

Operating expense will increase by the inflation and one half of the growth.

Data processing expense will increase by 1.25 times the inflation rate.

Manual expense will increase at the inflation rate plus twice the growth rate.

These assumptions do not take into account any major expansion which may be planned. Since large market expansion, new products, and opening new divisions, are variables onto themselves they must be excluded.

	LAST YEAR	THIS YEAR	NEXT YEAR
Estimated Inflation Rate		4.5%	5.5%
Estimated Revenue Increase		20.0%	15.0%
Projected Gross Revenue	38,500,000	46,200,000	53,130,000
Operating Expense	-26,750,500	-30,629,323	-34,611,134
Data Processing Expense	-550,000	-580,938	-620,877
Actual Gross Profit	11,199,500	14,989,740	17,897,989
Add Back Data Processing Expense	+550,000	+580,938	+620,877
Subtract Manual Process Expense	-1,007,995	-1,456,553	-2,104,720
Adjusted Gross Profit	10,741,505	14,114,125	16,414,146
Data Processing Contribution	457,995	875,615	1,483,843
Percentage of Dependency	4.1%	5.8%	8.2%

In the example we can see that while the cost of electronic data processing will increase by $70,877 between last year and next year, the cost of manual data processing increases by $1,096,725. This seems like an extreme variance. If the business has grown that much, why have data processing expense remained so low? Surely there will have to be increases in computer horsepower to handle the additional demand.

The example illustrates an income statement. If improved computer equipment and software are purchased to accommodate the demand, they will appear on the balance sheet. Manual labor is always an expense.

Having completed this exercise you should be able to form some sound conclusions. If the computer has been demonstrated to be a profit center, then the need for contingency planning is obvious. If the computer system were unavailable for an extended period, there would be a definite loss of profit. Profit and time are of the same kin. Neither can be recovered once lost. If you did not make the money this month you will not make up for it next month.

Human Resources commitment and understanding is the next stage to successful contingency planning. It is estimated that one million people are unemployed as a result of hurricane Hugo. A percentage of those left jobless because there employers no longer exist are not victims of the weather, but of their employers inability to react to the disaster situation. Regardless of the size of the business, contingency planning is needed to protect the livelihood of not only the enterprise, but that of the people who depend on the business for income. The work force must be made aware of this at the beginning of the recovery planning process. If they are not informed about the need to completely document what they do and how they do it, the planning process will appear to be an investigation rather than good business.

With any complex function such as preparing, processing and interpreting billions of bits of information, the most likely place for failure is the area with the most moving parts.

Key personnel turnover can be as disruptive to information flow as any natural disaster. Every company becomes dependent on the intuition of people. It is not possible for anyone to instill years of experience upon new recruits in the time period which most companies allot for training. The ability of people to adapt and make decisions in any situation makes an organization strong. Computer based systems, at present, do not adapt. If they did, business would not spend billions of dollars each year on customized software systems.

Contingency planning must recognize that the company is just as dependent on human intuition as on computer speed. The contingency planning process identifies what electronic functions are critical. It must also account for those functions which would not be possible if the personnel responsible were not available.

Relating this to the conventional attitude toward disaster recovery, it is easy to see how the hardware only approach is destined to fail. In almost every case the individuals responsible for preparing data and interpreting information work outside of the data processing department. Events such as labor disputes, health epidemics, employee turnover, stress and apathy can disrupt normal data processing with as much impact as computer down time.

Stress has been described as the silent killer. This is not only true of people, but can be equally applied to business. Dependency on key personnel is caused by an uneven distribution of responsibility and a lack of employee cross training.

Many of the duties performed by key personnel are not formally assigned to them. The nature of people who thrive on stress is to assume more responsibility, simply because they recognize the need to get something done. Eventually these individuals succumb to the health hazards of stress. When that happens, the business suffers along with them.

When responsibilities are assumed, management begins to lose touch with the way the organization operates. Decisions begin to be made at lower and lower levels. Eventually the management is replaced.

Granted this is not true of all companies, but the effects of stress have a relationship to contingency planning which is far more common. The stress felt by key individuals can manifest itself in a number of ways. We all know someone who on Friday at five pm is left trying to fix a problem caused on Tuesday. The weekend processing can not begin until the problem is corrected. If the problem is not solved by eight am Monday, then the processing for the new week can not begin.

When this is a rare incident it may not be important. For many new computer installations and in companies where responsibilities are assumed regularly, this example may be the rule rather than the exception. This type of problem fuels itself. As responsibility increases without the support and knowledge of upper management, the ability to communicate with management decreases.

Eventually something has to give. I QUIT is the usual reaction. This solves the problem for the individual, but leaves the business with a key man vacancy. Filling that vacancy will be costly. Without documented descriptions of the duties to be performed, the search for a replacement can not even begin. Once the position is filled, the learning curve will be extensive. Not only will the new employee need time to learn the business, their first assignment will be to figure out exactly what it is they are suppose to do. The result is financial losses to the business.

In a worst case situation, stress and a lack of recognition can result in extensive losses to the business. As responsibility increases, the demand for specialized knowledge of the key person increases. Eventually forward progress stops. More time is spent explaining than doing. This can move stress to a level of hostility. It may become necessary to let the key person go. It may even result in all out sabotage.

Contingency planning must prepare the business to deal with personnel dependency. It must identify not only the individuals who are key, but document the responsibilities which make them key. The number one reason for key man dependency is the lack of cross training. Dependency can be monitored and controlled by the

SUMMARY OF HUMAN RESOURCES AND CONTINGENCY PLANNING

People are as important to disaster recovery as computer hardware.

Contingency planning protects employment as well as profit.

Contingency planning must document who, what, why, how and when for all critical functions whether they are performed by computers or people.

Uncontrolled key personnel dependency is as dangerous as lighting.

Key personnel dependency can be controlled with cross training.

Contingency planning must illustrate the status of cross training and provide a method of monitoring key personnel dependency.

distribution of responsibility, accomplished through cross training. The more people that know how to do something, the more likely it will be done right.

Without total cross training, the key individuals will be the only hope of survival in a disaster situation. They are key, because of their ability to keep critical processing on schedule. Think of what happens when the disaster involves the lives of key personnel.

Throughout this reading you have seen the terms contingency planning and disaster recovery planning. It is now time to draw a line between these two separate items. Contingency planning is the on-going process of documenting and analyzing how a business functions.

Contingency planning, like business planning, never ends. It is a process, not a project. Keeping the contingency document up to date is critical. These documents serve as the back bone of the disaster recovery process. A change in one segment of the organization may go unnoticed in another. Yet, the overall ability to react in a crisis situation may be diminished due to that change. As the market for your goods and services changes, so will your approach to marketing. When interest rates change, so will your plans for expansion.

When a disaster occurs, disaster recovery is the mechanical process of executing the steps outlined in the contingency plan. It is therefore obvious that disaster recovery is dependent upon contingency planning. This becomes self evident when we explore the methods of disaster recovery available.

If disaster recovery were as simple as some believe it to be, you would not be reading this. Rule one of disaster recovery is compliance with the contingency plan. Having determined how much down time is available in the processing schedule and what the degree of dependency on information flow is, we must now seek out a disaster recovery method which fits these needs.

The following describes the opinions of most data processing professionals regarding the elements of disaster recovery; Perform and validate back-ups of system data regularly. Store back-up media at a remote location. Keep the computer environment under control. Restrict access to the hardware, software and data. Arrange for alternate hardware if the machinery is destroyed.

These are certainly important to disaster recovery, but, in reality they are things you are suppose to do whether or not you have a contingency plan. They are all part of disaster prevention not survival. Do not fault data processing professionals for this attitude. Without the commitment of management and a complete contingency document, it is usually the best they can do. Data processing management has always understood the growing dependency of business on computer resources.

The type and method of access to alternate hardware resources are defined in the contingency plan. When looking for a commercial supplier of these services it is best to prepare a list of requirements. Since we are dealing with time and capacity as the major factors in critical processing, they serve as the starting point.

For data processing, capacity is the easiest element to define. Capacity in disaster recovery means the minimum hardware configuration required to support the critical processing schedule. This means only the critical processing. It does not include the functions performed for convenience or software development (unless that is your business).

The cost of alternate resources in many cases is based on capacity. There is no need to pay for more than you need. By using the contingency plan that has identified critical processing, the data processing department can easily determine what the minimum requirement is.

The most important time element in preparing a resource requirement is allowable down time. This factor alone will determine the necessary delivery method of the alternate hardware.

There are readily available alternatives for hardware access. We will look at each and explain the pros and cons associated with them. In each case we will identify how the contingency plan helps in selecting which single option or which combination of options is best for you.

Simplest of all disaster recovery methods are emergency manual processing procedures. In cases of low computer dependency in key business areas, a well practiced program of using preprinted source and output documents may be sufficient. Unfortunately, the designers of computer software do not take disaster recovery into account. There main objective is to reduce the amount of manual work. This creates the situation discussed in the human resources section regarding distribution of responsibility. The more responsibility given the computer the more difficult the recovery process.

For those who have a high degree of computer dependency the benefits of good manual recovery procedures should not be over looked. The ability to respond to a disaster situation with manual processing increases the amount of recovery time available. When end users are capable of managing without the aid of the computer or the computer support staff the recovery team can function with less distraction.

The contingency planning process includes an analysis of both source and output documents. The input for each critical process is identified along with the method of data collection. Source documents for critical functions should be the first examined. Minor improvements and adjustments may make these more usable in a crisis. Special consideration should be given to the function of data entry from these forms. If the efficiency of data entry can be improved there will be a double return. Regular data entry will take less time and may become more accurate. In a crisis the data recovery time can be reduced.

Improving source documentation may increase the time required prepare input data. This additional time is offset by improved processing as illustrated in Fig 5.

Fig. 5.

Additional Seconds per Transaction Required to Prepare Source Data	+ 10
Reduced Seconds of Data Entry per Transaction	- 2
Reduced Editing and Correction Seconds per Transaction	- 12
Net Reduction in Transaction Processing Time	- 4
Total Transaction per Month	8,000
Total Hours Saved Monthly	9

Regardless of the disaster recovery method used you must employ manual recovery procedures for critical processing. The success or failure of the recovery process can depend completely on how well things get done without access to a computer.

The most common recovery method is a mutual site or handshake agreement. This is an informal arrangement between two or more compatible computer site locations. The basic principal is that if one site is unable to access their own system, the other will allow them use to theirs. The popularity of these agreements stems not from the practicality of them, but from the lack of upper management commitment. Without a contingency plan or budget, it is the only alternative available to many data processing managers.

This does not mean that they will not work. In companies with multiple site locations of similar configurations, that are under some form of centralized control, the mutual back-up agreement may be the best. Especially when each site performs the same critical processing. The main consideration must be capacity.

In situations where the site to be protected is the only site controlled by the company, it is more difficult to determine if the mutual agreement is feasible. Finding a partner with the proper hardware resource and with sufficient allowable down time will be the first step.

For the mutual agreement to be truly mutual it must take into account that once Company A is allowed to use the Company B computer, Company B has been placed into a potential disaster situation. Rarely will a business maintain an excess of computer resources large enough to support two companies. Due to this the Company B is placed at risk through the interruption in their own processing cycle.

MUTUAL AGREEMENT IMPLEMENTATION

SUPPORT SITE - COMPANY B	SUPPORTED SITE - COMPANY A
1 Make a full back-up	1
2 Validate full back-up	2
3	3 System configuration changes
4	4 Load system from last back-up
5	5 Perform catch up processing
6	6 Perform current processing
7	7 Make a full back-up
8	8 Validate full back-up
9 Re-configure system if needed	9
10 Load system from back-up	10

This process must be repeated each time the system is used.

To make room for Company A it will be necessary to first back-up the entire system. Company A can then install their system from their back-up. At the end of processing, Company A must back-up and Company B must restore the system to the point in time prior to system sharing.

Fig 6 lists the steps required for implementation of a Mutual Agreement. If the mutual agreement allows Company A to use the computer each night from 6pm to 6am, it would appear that twelve hours of processing time exists. The time needed for Company B to protect itself and for Company A to load the system must be subtracted. It is not uncommon for back-up, validate and reload processing to take as long as five hours. If this is true of our case, the actual productive processing time is reduced to two hours.

So far we have addressed the hardware capacity and time factors as they relate to mutual site agreements. There are major business and legal issues which also need consideration. Unless the person responsible for setting up the agreement is a principal of the company, it is unlikely that they have the authority to make a binding agreement which puts the company at risk. Even if the time and effort is made to prepare a formal contract for mutual back-up, it will probably not get past the corporate attorneys because of the potential risks.

The bottom line is that no CEO is going interrupt his critical processing while you run payroll.

MUTUAL AGREEMENT SUMMARY
Both parties must have sufficient capacity and allowable down time to support each other.
There must be a binding contract for mutual support which includes penalties for non-performance.
The agreement and procedures must be tested regularly.

Static cold-sites are complete computer environments. Static means the cold-site is stationary and that you will travel to it. They are ready to receive any computer hardware configuration which is compatible with the installed power and environmental controls.

The type of hardware you use and the equipment necessary to support it will determine the extent of services the cold site must provide. Before contracting cold site support the contingency plan must be consulted to see if a cold site is the correct option.

The cold site option instantly brings allowable down time into play. Availability of the required computer hardware, its delivery, and the time it takes to setup must be compared to the allowable down time. A portion of this time may be extended by the manual back-up procedures described earlier.

Mobile and portable cold-sites are available. Mobile indicates that the cold-site is on wheels. These are usually tractor trailers. Portable cold-sites are like modular homes and require special transportation. Both are equipped with power and environmental control devices designed for computer rooms. Some may be considered self contained and include electric power generators which operate on gas or diesel fuel. The floor space available is limited. The unit should be designed with a maximum computer configuration in mind.

The advantage to portable cold-sites is that they can be air lifted to reduce delivery time. The cost of delivery and setup of either type is usually the responsibility of the client. If this method of computer environment is used a special ride r should be added to the data processing insurance policy which will cover the delivery expense.

When considering mobile/portable cold-sites the allowable down time becomes the main factor just as it did with static cold-sites. Delivery and setup time for both the cold-site and the computer hardware must be accounted for.

When the contingency plan indicates that a portable or mobile cold-site is a viable option, you must next consider their special requirements.

You must pick the location where the site will be delivered. Most people think that the location is obvious. It will be put in the parking lot near the existing data center. That assumes that you will have access to the area around the building . If the parking lot is selected the location in the lot

SPECIAL CONSIDERATIONS FOR MOBILE/PORTABLE SITES
Additional insurance to cover the cost of delivery and setup
Selection of a site location at least five hundred feet from any buildings
Include the cost of site preparation when comparing to other options
Predetermine where computer users will be re-located
Cold-sites assume that you are replacing the computer hardware

should be the most remote possible. A rule of thumb is at lease five hundred feet from any buildings. If that is not possible, select a location elsewhere on the property outside the five hundred foot radius.

Once the location is selected it must be prepared. If necessary, the ground must be leveled. Adequate power for the computer hardware must be put in place and include a quick hookup junction compatible with that of the mobile/portable cold-site.

Telecommunications for both voice and data must also be put in place. Both power and communications must be independent of the main service to the existing data center.

Now, you must determine where the computer users will be located and what method of connection to the computer system will be used. Do not assume that they will be able to access any of the existing buildings.

All cold-site options make the assumption that the computer equipment is going to be replaced. This may be true, but in a disaster which makes the computer inaccessible without destroying hardware, the implementation and expense of a cold-site option is not practical.

Use of any type cold-site facility can be divided into two conditions. If the computer hardware to be supported is readily available for short time periods, and allowable down time is sufficient to wait for delivery and setup, the cold-site may provide adequate coverage. If this is not true the cold-site must be viewed as a long term support method.

Most important is the degree of disaster which must be sustained in order to implement the recovery plan. If you have insured yourself for the cost of implementation, the degree of damage will be dictated by your insurance carrier. Only all out destruction will probably qualify. If your down time is caused by an environmental problem, you would not activate the recovery process due to the expense. In this case you have actually been preparing and paying for a recovery support method you can not use.

Cold-sites tend to give the same sense of false security as business continuation insurance described in the conventional contingency planning chapter. You may have made all the necessary provisions for reacting to a disaster situation but you can not test them. It would not make sense to have the cold-site made ready with the required computer equipment because of the cost. Cold-sites are best used in conjunction with other recovery options as described later.

Mobile and portable hot-sites are also available. These are the same as portable cold-sites except they contain the required computer hardware to make them operational. They must be given the same consideration and preparation as the portable cold-site. The advantage here is one delivery time for both computer hardware and environment without the expense of replacing the computer equipment. If delivery can be accomplished within the allowable down time a mobile/portable hot-site may be the correct option. Even if the contingency plan indicates that a mobile/portable hot-site is a workable solution. There are two main problems with them.

First is testing. The cost of actually performing a full test, including the delivery of the hot-site, is prohibitive. Minimum testing would require that you go to the portable hot-site. It is necessary therefore, that supplier of the site to keep it operational.

Secondly, and more importantly, is the degree of disaster which must be sustained in order to implement the recovery plan. If you have insured yourself for the cost of implementation, the degree of damage will be dictated by your insurance carrier. Only all out destruction will probably qualify. If your down time is caused by an environmental problem, you would not activate the recovery process due to the expense. In this case you have actually been preparing and paying for a recovery support method you can not use.

Remote access hot-sites are made possible by the advances in data communications. This recovery approach is the most popular in large network environments. The decision to use this method in a closed network situation must be made using the same guidelines as the mutual site agreement.

Remote access means that the computer users are located away from the computer hardware. When a commercial supplier of remote access hot-sites is contracted for services, the contingency plan must approve of the dependence on outside expertise for computer room operations. The most common remote access facilities are time sharing companies. This means you may have limited access to the computer just like mutual agreements.

This method may have an inherent flaw. It may depend on a third party for success or failure. Unless you have your own telecommunications network you will be at the mercy of someone else. Another consideration is the expense of telecommunication equipment which may be needed and the relocation of computer users during the disaster.

A static hot-site is a complete computer installation dedicated to disaster recovery. The key word here is dedicated. If the facility is used for any other regular purpose such as payroll services it can be called a true hot-site.

The decision to contract for static hot-site support is made based on total computer dependency and allowable down time. The static hot-site provides nearly immediate access, depending on travel time. Most static hot-sites also support remote access. In some cases they may provided enough space to relocate an entire business if necessary.

The biggest drawback for static hot-sites is cost. Subscriptions to hot-site facilities are the second most costly recovery method next to redundant systems. Fees can range from all inclusive annual fees which cover testing and a fixed number of days the facility will be made available, to annual fees, plus daily fees, plus testing fees. When comparing bids for hot-site support it is important to match quotes, service for service.

A static hot-site contract should have regular testing built-in. This serves two purposes. It shows flaws in the recovery plan which can then be corrected and guarantee that the hot-site can support your processing.

Redundant computer configurations are for companies which can afford them. This method involves the purchase and setup of duplicate computer operations which become a company owned hot-site. The problem is that they do not remain a hot-site for long. If a company is so computer dependent that the cost of redundancy can be justified, they probably have a need for unlimited computer resources. The hot-site system will eventually become a production machine.

Combining different recovery methods must be considered if the total protection of information processing is the objective. Transportable cold-sites can be combined with any of the methods which provide access to computer equipment. The combination of a static hot-site and a portable or re-locatable cold-site provides the widest range of coverage.

When a short term interruption occurs the static hot-site provides continued computer access. If the interruption is long term, the hot-site keeps processing on schedule while the cold-site and new computer equipment are delivered and setup. When the cold-site is operational the hot-site is vacated. Once the original computer facility is ready to receive equipment, processing can move back to the hot-site while installation takes place.

Since the Contingency Plan will involve all aspects of data processing it will, as one of its objectives, document how information is being used. It will illustrate how we can improve on its use. Make something more efficient and it will cost less. The greatest cost savings can be realized in the area of greatest expense. Human Resources.

For the contingency plan to be effective, it must document who is responsible for each element of critical processing. This documentation becomes a guide to internal promotion. When an individual leaves the company, we can easily identify a path of promotion by reviewing the cross training elements of the plan. Decisions can be made with facts. As stated earlier, the disaster recovery procedure used will depend on the disaster situation. No amount of planning will prepare you for all situations.

If the office building is destroyed by fire, hardware replacement becomes a key factor.

If the office is closed by the EPA for asbestos removal, temporary relocation is the main issue.

If a disgruntled employee damages computer equipment, repair time becomes the deciding factor.

If a key employee is killed, personnel dependency is the issue.

It is easy to play a game of what if when considering the number of possible events which can and will upset the normal routine of daily business. What is important is being prepared to act.

The contingency plan must serve as the source for assessing damage and the extent of losses. Only by knowing the true impact of the failure wave can steps be taken to catch up lost time and prevent continued losses. In order for the contingency plan provide you with the information necessary to prepare those steps it is obvious that the plan must be comprehensive. The following outline illustrates some of the key information points which should be included.

I. COMPUTER HARDWARE
- A. VENDOR
  1. SALES SUPPORT
  2. TECHNICAL SUPPORT
  3. REPAIRS & MAINTENANCE
- B. PERSONNEL
  1. EQUIPMENT MANAGEMENT
  2. RESOURCE MANAGEMENT
  3. OPERATIONS MANAGEMENT
- C. OPERATIONS
  1. BACK-UP PROCEDURES
  2. VALIDATE PROCEDURES
  3. MEDIA LIBRARY
- E. OPERATING ENVIRONMENT
  1. OPERATING SYSTEM LEVEL
  2. NETWORK ENVIRONMENT
  3. SYSTEM CAPACITIES
- F. EQUIPMENT
  1. SERIAL NUMBERS
  2. REPLACEMENT DATA
  3. VENDORS
- G. INSURANCE
  1. AGENT
  2. POLICY
  3. EXTENT OF COVERAGE

II. COMPUTER SITE
- A. PHYSICAL LOCATION
  1. NEIGHBORHOOD
  2. BUILDING STRUCTURE
  3. COMPUTER LOCATION
- B. PHYSICAL SECURITY
  1. BUILDING PERIMETER
  2. BUILDING ACCESS
  3. COMPUTER ROOM ACCESS
- C. PREVENTATIVE MEASURES
  1. FIRE PROTECTION
  2. WATER PROTECTION
  3. ATMOSPHERE
- D. ENVIRONMENT CONTROL
  1. UTILITY COMPANIES
  2. AIR CONDITIONING
  3. POWER CONDITIONING

III. HUMAN RESOURCES
- A. IDENTIFICATION
  1. HOME ADDRESS
  2. EMERGENCY TELEPHONE
  3. POSITION IN ORGANIZATION
- B. WORK ENVIRONMENT
  1. WORKING HOURS
  2. OFFICE LOCATION
  3. DEPARTMENT
- C. PRIMARY DUTIES
  1. APPLICATION ASSOCIATION
  2. DESCRIPTION
  3. DOCUMENTATION, BACK-UP
- D. SECONDARY DUTIES
  1. APPLICATION ASSOCIATION
  2. DESCRIPTION
  3. DOCUMENTATION, BACK-UP
- E. WORK ROUTINE
  1. TIME PERIOD
  2. APPLICATION
  3. COMMAND REFERENCE
- F. BACK-UP RESPONSIBILITIES
  1. FOR WHOM
  2. TO DO WHAT
  3. DOCUMENTATION
- G. SECURITY
  1. BUILDING ACCESS
  2. COMPUTER ROOM ACCESS
  3. SYSTEM ACCESS

IV. APPLICATION GROUPS

- A. IDENTIFICATION
  - 1. TYPE
  - 2. SUBSET
  - 3. RELEASE LEVEL
- B. DEFINITION
  - 1. COMPUTERIZED
  - 2. MANUAL
  - 3. PURCHASED OR IN-HOUSE
- C. PERSONNEL
  - 1. DATA ADMINISTRATION
  - 2. TECHNICAL SUPPORT
  - 3. VENDOR SUPPORT
- D. STATUS
  - 1. DOCUMENTATION
  - 2. RESOURCES
  - 3. SUPPORT
- E. CUSTOMIZED PROGRAMMING
  - 1. WHICH AND WHEN
  - 2. BY WHOM
  - 3. FOR WHAT REASON
- F. CUSTOMIZED DATA FILES
  - 1. WHICH AND WHEN
  - 2. BY WHOM
  - 3. FOR WHAT REASON
- G. RELATIONSHIPS
  - 1. DEPENDENT ON
  - 2. RELATED TO
  - 3. ASSOCIATED WITH
- F. PROGRAM AND DATA FILES
  - 1. IDENTIFICATION
  - 2. STATUS
  - 3. BACK-UP FREQUENCY
- H. IN-PUT AND OUT-PUT
  - 1. SOURCE DOCUMENTS
  - 2. HARD COPY DOCUMENTS
  - 3. SECURITY AND CONSUMPTION
- J. PROCESSING SCHEDULE
  - 1. TIME PERIOD
  - 2. COMMAND REFERENCE
  - 3. PERSONNEL
- I. SPECIAL EQUIPMENT
  - 1. DESCRIPTION
  - 2. DEGREE OF NEED
  - 3. VENDOR

V. PERSONNEL DEPENDENCY

- A. ANALYSIS
  - 1. RESPONSIBILITIES
  - 2. POSITION IN ORGANIZATION
  - 3. BACK-UP, DOCUMENTATION
- B. DEPENDENCY
  - 1. KEY PERSONNEL
  - 2. DUTIES DISTRIBUTION
  - 3. VALIDATION OF DATA
- C. RISK MANAGEMENT
  - 1. CROSS TRAINING
  - 2. SKILLS ANALYSIS
  - 3. PROMOTION PATH

VI. APPLICATION DEPENDENCY

- A. ANALYSIS
  - 1. COMPUTER RESOURCES
  - 2. HUMAN RESOURCE
  - 3. BACK-UP, DOCUMENTATION
- B. DEPENDENCY
  - 1. OVER ALL STATUS
  - 2. CUSTOM CHANGES
  - 3. VALIDATION OF DATA
- C. RISK MANAGEMENT
  - 1. CROSS TRAINING
  - 2. DOCUMENTATION
  - 3. PROCESS IMPROVEMENTS

VII. VENDORS AND SUPPLIERS

A. IDENTIFICATION

1. REPRESENTATIVES
2. CONTACTS
3. CLASSIFICATION

B. GOODS AND SERVICES

1. HARDWARE
2. SOFTWARE
3. SUPPLIES

C. SECURITY

1. SITE ACCESS
2. COMPUTER ROOM ACCESS
3. COMPUTER SYSTEM ACCESS

VIII. PROCESSING TIME TABLE

A. APPLICATION LEVEL

1. FREQUENCY
2. START TIME
3. DURATION

B. SYSTEM LEVEL

1. TOTAL PROCESSING
2. ALLOWABLE DOWN TIME
3. SCHEDULE CONTROL

IX. RECOVERY PROCEDURES

PHASE 1: ASSESSMENT OF SITUATION

PHASE 2: INITIAL RECOVERY MEETING

PHASE 3: PREPARATION FOR RECOVERY

PHASE 4: RECOVERY IMPLEMENTATION

45 Training Tips: A Tip-A-Minute

Tom Elliott
Distribution Resources Company
6061 S. Willow Drive, Suite 100
Englewood, Colorado 80111
(303) 889-4500

The following tips can be applied to a variety of situations, therefore the words involved should be used interchangeably to fit the occasion. The words "presenter, trainer, instructor", etc. are aimed at the individual that is identified as the leader of the discussion. The words "student, participant, learner", etc. are the people that are supposed to benefit directly from the golden words of the others.

There are three kinds of training: Not at all (we don't have the time or the resources), poorly developed (we don't have the time or the resources) and effective (we took the time and found the resources). It is but one link in the overall sales, service and follow-up scheme of things. However, training tends to fall into one of the first two categories because it is easy to overlook or set aside when time constraints take over practicality. Everyone realizes that training is important, but.....

Effective training is both a science and an art. As a science there are some tried and true methods and procedures that can lead you to giving a successful presentation. The personality of the instructor fulfills the art aspect. Training can be the difference between a successful implementation and a dismal failure. It can mean a more positive effect on a company's bottom line SOONER rather than later. Therefore, good, sound and effective training should be the goal in the creation of any training course.

Today we are going to review a series of ideas that everyone knows and some even remember. It is the goal of this session to give you just one idea you can take with you and use in your own environment. Just as it is important for a listener to find the time and resources to attend a training session, so must we take the time and resources to give them their money's worth.

****
*   THE COURSE   *
****

1. DEVELOP COURSE GOALS

If distinct goals cannot be defined the course should never be offered. A course should have a definite starting place and a definite ending place.

2. MEET THE GOALS OF THE COURSE

Although this sounds simple enough keep in mind that the training course must contain an atmosphere of learning. This means a sound two-way communications situation. Some students and some instructors like to talk...and talk...and talk. This means a constant adjustment of material presentation. If you are unable to get through the entire course and present all of the critical topics you have shortchanged the student...and yourself. To keep this from happening, have mileposts that coordinate the materials and the time. Halfway through the course in time you should have a good idea of where you should be in the presentation of materials. A good way of creating the mileposts is to know where you would like to be at every break and at the end of each day.

3. ASSESS THE CLASS TIMING

Each class will be different because the students are different and you are different from day to day. Some classes will have inquisitive students and will ask questions frequently. You will find these are typically the most rewarding but they will also take the most time. Other classes will contain people that feel uncomfortable asking questions during class and you will cover the material much quicker. If you want the class to end at 5:00 pm recognize some classes may go until 6:00 pm and some may end at 4:00 pm. It is best to priortize the importance of the topics and make sure you cover the high priority items. The very low priority items might be used as filler if needed. If filler is not needed don't worry about not covering a topic. To put it bluntly...BE FLEXIBLE.

4. TRAINING DESIGN IS CRITICAL

When we are putting a presentation together we should be spending more of it on the design and flow of the topic. A good speaker can be better if the design is solid but a good speaker will be only good if the design requires them to work harder in organizing their material. If the presentation is a repetitive one the speaker should critique the design immediately after the delivery to identify the rough spots and to take corrective action in the design.

5. KNOW THE STUDENTS

It is important to know the mixture of the class. Is everyone of the same background and experience? If so, you can move the class along faster. Be aware that no software can be designed to be all things to all people. That means there will be people in the class that don't agree with your solution. You must make them aware that is alright, but they are being taught existing solutions and not "what if's..." or "it should be's...". If the mixture is not homogeneous, it will take more time to cover the same materials. Allow for it.

6. RESTRICT YOUR COMPUTER BUZZWORDS

Unless you are technical by nature and your students are technical by nature the use of buzzwords are probable going to be detrimental to the discussion. If you need to use buzzwords, you must teach them what those words mean. Very few people are going to accept technical jargon by asking for a clarification at the time they need to understand what the word means. Typically they are embarrassed to ask for the meaning of a word that must be second nature to you. Use their terminology or teach the use of any special buzzwords in your presentation.

7. CONTROL THE ENVIRONMENT

Remember, you are the teacher and the students are looking to you for leadership. All things hinge on your ability to recognize the situation and respond to it, keeping in mind the teacher/student relationship priorities. There are a lot of tricks to keeping control and your experience will be your best friend. Speaking of "friends", being a teacher must take priority over being a friend.

8. UNDERSTAND THE POWER OF PARTICIPATION

Nothing keeps the attention of the student better than knowing they will be part of the exercise/class. Group participation, such as gaming, will tend to lessen the stress of the participation if that is desired. Role playing, taking turns giving answers, etc. will put more emphasis on the individual.

9. USE THE THREE-MINUTE RULE

A lot can be accomplished in three minutes, including getting the listeners to participate in the delivery of the message. A "show of hands" and counting the results, asking a question (one) and getting an answer (one) certainly take less than three minutes and can lead the presenter into the appropriate level of material. You can even have people write something down and share it with a neighbor in less than three minutes. They can even present it to the group in less than three minutes if you structure it correctly.

10. MAKE USER OF AVAILABLE MEDIA

No matter what the media, be sure your visuals are sound in their creation. Think of the visual as your note cards. If you know your topic well the visuals will serve two purposes: they will focus the student's attention where you want it and it will feed you a key for further discussion. Good rules might be:

A. Keep the number of words below 30
B. Use color combinations whenever possible
C. All words must be readable from all parts of the room
D. Pictures and other graphics are very effective if they have meaning
E. Charts are very effective if they are simple and direct and do not have extraneous information on them.

11. CREATE GOOD HANDOUTS

Handouts are generally very helpful to the student if they are dealt with properly. They can also be a detriment if their use is not controlled. If they are going to be thumbing through the handouts while you are making a point, they will miss the point and they may make someone else miss the point. Be aware that handouts can allow the student to make better notes but they might also be used for a student to look ahead when you don't want them to. Generally speaking, handouts are good. Be aware of the pitfalls in advance and decide how you want to deal with them.

12. DETERMINE WHAT, IF ANY, DETAIL INFORMATION MIGHT BE OF BENEFIT

In some instances detailed writeups and data sheets might be helpful to the student. Have them ready to hand out at the end of the session, even at the break times. Do not hand them out in advance! At the appropriate time in the presentation make reference of the handout so the student can make a note of it and determine if they want a copy. Try not to take class time to pass out any literature or information. It tends to keep the class from determining a pace.

13. ARE YOU RIGHT-HANDED OR LEFT-HANDED?

Each individual will tend to pay more attention to one side of the room or the other. In some cases, that will be due to the way the room is laid out. Be aware of where you are the most comfortable and work hard to cover the other side. If you have a chance, you might want to put the people you want to impress the most on the side you tend to work toward. You must know if you teach to the right (for example) you will tend to ignore the people on the left.

14. TABLE ARRANGEMENT

The physical arrangement of the tables will have some effect on some of the potential problems you will have to deal with. If the students are in a U-shaped environment they can talk to each other easier and get to acknowledge all of the other students because they can see them. It also tends to limit the number of people you can teach at a time. Rows of students allow you to work with more people at a time but those in the front of the room must turn around if they want to see who is talking. If that is your situation and if you know the students, put the more lively ones in the back, forcing those in the front to participate more by turning around. Your eye contact will tend to cover the whole class better.

15. SEATING ARRANGEMENT

If you know the personalities of the students consider putting a "star" with a "non-star" only if you want to facilitate a "help" atmosphere. Otherwise the "non-star" might be intimidated by the "star", creating a problem. If you are using a screen put it in the front right-hand corner of the room at an angle. This gives you the best shot of everyone at least seeing the screen.

16. SCREEN ARRANGEMENT

To give the best viewing angle when a screen is being used, put it in the front right-hand corner of the room. If you are also using an overhead projector where you will be "working the table" (writing or working with information on the overhead) be aware of the table arrangement and your right/left handedness. Put the screen in the front corner where your physical being will be in the way the least!

Computer
Museum

17. KNOW THE SUBJECT

If you are going to impart knowledge you must have the knowledge to impart. You won't know all there is to know on the topic to make a presentation but you should be very well versed in the topic at hand. Be aware, however, that the customer may ask questions you have not thought of. If you don't know an answer say so...then find the answer and make sure everyone in the class is aware of that solution.

18. BE AWARE

Any concept that has good points can also have pitfalls. When you make a decision about the class in some fashion think of what could go wrong. If you are not surprised when things go wrong in the class you will be able to handle them better. The worst thing for an instructor is to be blindsided by a student and not be able to handle the situation. Whether the problem is a simple question you would never have thought about or it's the fact that someone in the room cannot see the board the more you are aware that something could happen will help in getting around the problem.

19. USE THE TEAM CONCEPT

A classroom is an opportunity for all to learn. The instructor will lead the discussion based upon a set of predetermined goals and materials. The student, however, has knowledge the instructor will never have unless the student imparts that information. The value of the instructor is how to get the information they have to the student and how to draw information from the student to build on their own base. The more knowledge is exchanged the more successful the presentation will be. Both groups must listen, talk and understand.

20. ATTENTION SPANS

If you are lecturing, 50 minutes of sitting is about all a rear-end can handle before it need to be moved around. If the student is working exercises on a terminal they become more intensive and can last twice that long. If you are lecturing, watch the eyes of your audience. When you see eyes drifting shut, it's not necessarily because you are boring (although that may be the case) it may just be that their "sit level of exhaustion" has just been reached. Our attention span will be shortened in the afternoon, with a particular problem in the 2-3 pm range. Take breaks to get the brain working again or give the students a change of pace, say from lecture to participation exercises.

21. TEACH WHAT IS IMPORTANT

Be careful of this one. A student figures that any topic you emphasize in class MUST be important, otherwise why would you be making a big deal of it? You must therefore NOT emphasize a topic that is not really important. Emphasize only those topics that meet your class goals and have a top priority. Downplay those topics that don't fit that definition or put them in later in the day.

22. TEACH AT THE LEVEL OF THE STUDENT

The student will be from a variety of backgrounds and intelligence levels. You must be able to put yourself in their position mentally before you can teach because you must understand their level of thinking. One day you may be making a presentation to corporate executives, interested in only the overall concept of the system functions and the relationship of those functions to their responsibilities. The next day you may have data entry clerks whose responsibility is much more narrow and intensive but no less important to the scope of the company.

23. MAKE THE STUDENT COMFORTABLE

If the student knows you are working to communicate at their level they will open up to you. Take the time to know names, positions, responsibilities, etc. At the very beginning of class use "ice-breaker" exercises to promote the student's being comfortable before class begins. If chairs, desk height, etc. do not promote physical comfort, be aware of a potential reduction of attention span.

24. TEACH THREE TIMES

At the outset of the presentation of a module (no matter how large or how small) tell the students what you will be teaching them, including the goals of the course. Then do the actual teaching. When completed with the module, tell them what you have just covered to reinforce the successful completion of the goals.

25. BE FLEXIBLE

Just about the time you think you have the world by the tail it will turn around and bite you. All of your planning will seem to go by the wayside. You must acknowledge Murphy and the chance that something will go wrong and be prepared to shift gears on the fly. The presentation must continue to be smooth. Don't lose your cool or get frustrated.

26. HAVE MEANINGFUL BREAKS

It has often been said that the ability of the brain to absorb information is directly related to the amount of time the rear end can sit still. Design your breaks at logical stopping points if at all possible. Always remember that there will be a new "start up" period when the group comes back. If you want to give the students a 15-minute break it will take 20-25 minutes before you have their attention again.

27. GIVE ODD TIMES TO START AND STOP

This may seem trivial but it works big time. If you want to begin a session at 8:30 am indicate the session will begin at 8:27 am. When you stop for a break give the students 12 minutes rather than 10 or 15. Have the students "synchronize their watches". This will bring the time to their attention. When the exact time arrives, begin or continue your presentation.

28. GAMING

You have to understand that learning can only take place when you have the attention of the learner. One very effective way of doing this is to create a "game" to put points across. Dividing the class into smaller groups will put everyone into the "competition". You might want to do a take-off on a popular TV game show such as "Jeopardy". Give points and prizes but make the questions relative to the topic at hand.

29. VIDEOTAPE THE STUDENTS

It is very popular to use the videotape these days to point out problems and to help correct them. When using videotape to critique a presentation we like to point the camera to the instructor and emphasize the little things; "Look at the camera more", "You looked at the camera too much", "Smile more", "You scratch your head too frequently", "Next time wear brighter clothes", etc. Although beneficial to a degree we should point the camera to the listeners and see if they are receiving any benefit from the MESSAGE, or are they yawning, looking at their watches, etc.

30. COMMUNICATE!

Strangely enough, most people think that training is being able to lecture. In fact, the most important part of teaching is having the ability to LISTEN. Converting what you hear to the product at hand is the difference between success and failure. You must know the problems and concerns of the student before you can show them how the product will work for them.

31. HONESTY IS THE BEST POLICY

Although this phrase has been used so often most of us don't take it to heart, it is still a very critical key to facilitate the training procedure. In order to do this, there is one thing that must be kept in mind: Very few problems are not solveable in one form or another. If a student asks you about a function you don't think is available, make sure you get much more information before you say "NO". Perhaps the question was not put well or the student has some predetermined answer in mind that does not fit. But if you can find out what their real problem is you might be able to define a work-around. If you know the problem and the solution is not available for that problem say so! Never offer the fact that programs can be written to handle the situation! That approach turns the student off. If they ask if a program can be written to solve the problem you should be able to answer affirmatively, assuming the data they need is stored in the computer somewhere. If a question is asked and you don't know the answer, respond that you don't know but you will find the answer and get back with them. Then do it! Stop then to take the time to make a note to yourself being sure you write down all the pertinent facts. Then follow up on it the first available opportunity. Your goal is to find the answer within ONE HOUR.

32. KNOW YOUR LIMITATIONS

Don't get yourself into hot water by trying to talk about something you are unsure of. It will eventually lead you to expose your stupidity.

33. KNOW YOUR BODY LANGUAGE

As you work with the class there are things you can do to improve the communications. If a student asks a question, take two or more steps toward them. This indicates you are listening intently and are sincerely interested in their inquiry. If the person happens to talk softly, repeat the question to the rest of the class before you answer it. Smile. That tends to make others smile and that make them relate to you better. Be aware that much of what you do tells the student if you want them to listen or not to listen.

34. MOVE AROUND

An extension of the body language concept you must make the class as interesting as you can. If you stand in one place with your hands at your side you will make the class boring. If you move and point and vary your speech level you can keep the attention of the student better.

35. WORK AT BEING YOURSELF

You have been selected as an instructor because of your personality and your ability to learn. You may have to work at being able to express yourself in front of large groups but that can come with effort and time. You are not "GOD" and no one expects you to be anything but prepared. Develop your own level of being comfortable and do not try to mimic other instructors. Only when you know yourself will you be able to relax in front of a crowd. Another way of saying this is to DEVELOP A LEVEL OF CONFIDENCE IN YOURSELF AND YOUR ABILITIES.

36. HAVE A THICK SKIN

Not everyone will agree with what you are trying to teach them. And they will tell you!

37. HAVE PATIENCE

The foremost quality of a good teacher is patience. People will push it to it's extremes. AT NO TIME SHOULD YOU ALLOW A STUDENT TO TAKE YOU PAST YOUR PATIENCE LEVEL. You need to develop techniques to keep you away from that possibility. If you do not have patience you will formulate defensive answers to a comment a student is making instead of listening and developing a positive solution.

38. HAVE FUN!

If you don't set the stage to enjoy the class, the students will have a bad time, less learning will take place and it will be a real drag. Have a good time, enjoy the opportunity to help, relax, share your knowledge and broaden your own experiences and information base.

39. PLATFORM SKILLS ARE NOT ENOUGH

Being able to stand up in front of a group and speak well is a talent but is NOT the primary requisite for imparting knowledge. Most speakers with this talent tend to enjoy speaking but can sometimes enjoy hearing themselves too much. The listener is the key ingredient and having them achieve the goals intended is the primary function of the speaker.

40. ENTHUSIASM HAS IT'S PLACE...BUT...

Enthusiastic speakers make the listener, and the speaker, feel good about what they are hearing. Remember, though, the listener must still learn! Stories, jokes, smiling, kidding the audience, etc., are great tools to get the listener to participate in the training but should be integrated into the message and used as tools, not as the means.

41. EVEN TECHNICAL PEOPLE CAN BE GOOD INSTRUCTORS!

It has often been said that technical people know their material but can never present it to someone else. If people learn some of the basic skills needed to be a presenter/instructor they, too, can be successful in passing on their knowledge. A person with any kind of background, technical or otherwise, is asked to present they will take one of two directions: if they have any presentation skills they will utilize them, else they will fall back onto the method of presenting they know best, the lecture.

42. REPITITION IS ONLY A LEARNING PROCESS

"Practice, Practice, Practice" is the best way to become perfect. Just as it does not pertain to other endeavors all of the time, neither does it pertain to training. Learning to play golf and learning to present are the same. If you don't learn the solid basics at the beginning you can only go so far toward attaining a high skill level. Just as you will find golfers with a high handicap that have been playing for 20 years so you will find instructors that are mediocre after 20 years.

43. LEARN FROM EXPERIENCED PRESENTERS, GOOD AND BAD

If you want to gain skills as a presenter you need to learn to critique all presenters, both good and bad ones. Not an easy task because you will tend to get caught up in the message from the good ones and turn off the bad ones. If you take your time to analyze what skills cause you to react to the speaker you can benefit greatly.

44. TAKE RATINGS WITH A LARGE GRAIN OF SALE

Quite frequently you are asked to rate a presenter. The method of rating can vary from the specific question/answer to the "What did you think?" kind of opportunity to respond. You are then asked to "drop them in the box on the way out". That gives you a grand total of about 30 seconds to rate a presenter. Feel good about receiving the message and you give short but rave reviews. Not pertain to you or not cover the special topics you came to hear about and you give non-positive reviews. Not completing the review tends to be the most frequent review and nobody knows what that means. When looking at your reviews, ignore the positive ones and concentrate on the negative ones. Those are the reviews that will teach you about your skills to deliver a message.

45. ENJOY THE BUTTERFLIES!

Many people will never be good presenters because they misinterpret their nervousness. They think they are nervous because they are unsure of their skills, of their topic, of their audience, of the time of day, etc. Butterflies are the result of adrenaline pumping to prepare you better mentally. Presenters that have little or no butterflies prior to their efforts will have flat performances.

"BUTTERFLIES IS GOOD"

Education: In-house, Out-house or CBT?

James F. Davis
Cognos Corporation
2301 E. Lamar Blvd., Suite 250
Arlington, TX 76006
(817) 649-1944

One of the most important and also the most difficult tasks that faces you as a technical manager each year is budgeting and planning for the education and training of your staff (should it be in-house, out-house or computer based training?). Why is training or education so important? The ability of your staff to learn and retain specific techniques and skills can directly affect your organization (staff morale, staff productivity, crisis management and your own performance measurement).

To be successful, an education plan has to meet four requirements: the training must be appropriate, the information must be correct and current, the student must retain the information and the training cost must fit into the budget. It is difficult to make sure all four requirements are met. Yet, would you be willing (or able) to sacrifice any one of them?

I am the Education Coordinator for Cognos Corporation's South/Central Area. My duties include the scheduling and co-ordination of PowerHouse education services for our customers in one third of the U.S., as well as South and Central America. I am responsible also for the certification of our instructors. Because of my position, I have the unique opportunity to observe many different approaches to the responsibility of education planning. Three ineffective approaches have become apparent.

Some managers dole out training as if it was part of their staff's compensation. While this approach is noble, it is often done in a haphazard fashion, with no long range planning. The training is approved as long as the course cost fits into the budget and the subject is somewhat related to the company's business or the person's specialty.

Other managers are very tight fisted with their training budget, reluctantly approving education only when there seems to be no other way for the employee to be trained. Again there is no long range planning involved - training is used as a quick fix for a particular situation.

Still others will not release their staff for training that lasts a full week. They contend that it is impossible to do without their employee for that length of time. If this line of reasoning is true, it is frightening. No employee should be so indispensable that you can not do without him or her for a week. This type of excuse also points to a manager's shortsightedness, a refusal to trade a week of an employee's time for years of benefit.

Actually, all three approaches are symptoms of a larger problem. Most managers do not feel qualified to make proper education decisions when faced with the myriad possibilities that are offered. There is only one way to approach this problem. The different types of education and training must be studied and understood before a responsible decision can be made.

When planning for education, there are four types of training to be considered: college education, professional classroom training, professional self paced training, and training developed in-house. College education that is applied toward a degree is usually at least partially funded by many employers. This type of education is rarely under the manager's direct control. Therefore, it will not be addressed in this paper.

The other types of education are what most managers must choose from. Professional classroom education, offered by most software and hardware vendors as well as many third party vendors, is one obvious possibility. Professional self paced education, provided in workbook, computer based and video formats, is also available for purchase. Education that is developed in-house is also an option. It can include both classroom and self-paced training.

This paper will explore the advantages and disadvantages of each type of education available. The cost of each type will be addressed separately from the advantages and disadvantages. Cost as a positive or negative factor is only relevant under specific conditions. All types of training must be considered together. The potential number of people to be trained must also be included to achieve a fair comparison. Therefore, cost will be addressed after the positive and negative aspects of all training programs have been covered.

PROFESSIONAL CLASSROOM TRAINING

Professional classroom training is offered by many sources. In most cases it is available in one of three formats: public, on-site and customized training. The enrollment is usually limited to a small number of students. The standard approach is a mixture of lecture, discussion and workshops.

Public training is usually offered at the vendor's central training facility. Anyone who pays the fee is allowed to attend. It is rarely limited to the employees of a specific company.

On-site training is usually the same material that is offered in the public classes. However, it is taught for a specific company, at their chosen location (usually "on-site," at their place of business).

Custom classroom training is often an extension of standard public training, offered at additional cost. Customizing may be as simple as replacing the examples in a standard course with examples provided by the company. Customizing may be as complex as designing a total course that meets the specific needs of the company. In most cases, the vendor that develops the training delivers the training. However, it is possible to purchase courseware from some companies and use one of your own employees to deliver the training.

One other type of public training needs to be addressed. Seminars are often offered by the same companies that provide professional classroom training. A seminar differs from classroom training in several aspects. A seminar is usually open to 25 or more attendees. The size of the seminar often discourages questions and discussion and makes workshops impossible. A large amount of material is usually presented in a short amount of time. These factors can greatly affect the attendee's retention rate. Seminars present a wonderful opportunity to get a good overview, or pick up specialized tidbits. However, they are not designed to give a comprehensive understanding of any subject.

Advantages of Professional Classroom Training

There are many positive aspects to professional classroom training. Most are provided by professional trainers and developers. There are also some specific pluses associated exclusively with public, on-site and custom training.

Professional Trainers

Professional classroom training's greatest strength is reflected in its name, in that it is developed and delivered by professionals. Because of this, you should expect a high level of quality in the course materials and the way that they are delivered.

Most organizations that offer professional classroom training require their instructors to complete specific requirements before they are allowed to deliver a course. Even after the instructor is certified, most companies will "spot check" their instructors to make sure that they are employing good teaching techniques. If the course is offered by the vendor, you also should expect the instructor to be up to date on any new developments.

Instructor certification requirements often include some kind of "train the trainer" course that concentrates on adult teaching techniques. These techniques help to insure the highest retention rate of the material covered. Further, most vendors also require their instructors be "certified" in each specific course that they will teach. This may only require that the instructor possess a certain level of technical knowledge. It may also include some of the following: observation, co-teaching with a senior instructor and solo teaching. Each step that is required ensures a greater level of integrity.

<u>Professional Course Developers</u>

There are many advantages to a course that has been developed by professionals. These include course approach and flow, content reinforcement, iterative development, up-to-date courseware (especially when offered by the vendor) and course notebooks.

When a course is developed by professionals, the approach and flow are given almost as much attention as the actual content. It is important that the course approach the subject from the best possible angle. Which material is basic foundation material, which is more advanced? It is also important to ensure that the course is developed so that it flows properly from one topic to the next. Course flow sounds trivial but actually is extremely important for the course's overall effectiveness. For example, the modules must be planned so that major break points (lunch, day's end) fall at an appropriate place. Likewise, more difficult material should be scheduled early in the teaching day so that it is presented when the students are fresh.

Content reinforcement is an important aspect of any course. It is easy to follow the instructor while he or she works through examples at the front of the class. The real learning occurs when the students must use the new concepts themselves. The professional course developer will provide tests for understanding. Specific topic questions, case studies and class discussions all help the instructor ensure that the material that has been presented is actually understood. In situations where equipment or computer language syntax is involved, hands-on workshops should be provided to let all the students get a "feel" for the material that is being presented.

Iterative development is another benefit of the professionally developed course. A new course is often implemented with a "beta teach." After the work on the course is complete, it is "taught" to the instructors who will be teaching the course. Their input is used to enhance the course before it is released. As the course is taught, suggestions from the instructors and students are incorporated into subsequent presentations and newer versions. With each new version, the course is improved with better, "real world" examples and techniques.

A professionally developed course should always be up-to-date. This is especially true if it is offered by the vendor who produces the hardware or software being discussed. With access to the research and development teams, the professional developer can often incorporate new developments into courseware for a coordinated release.

Professionally developed classroom education always provides a quality student workbook. This workbook is an essential aid for good retention of the material. An important aspect of professionally developed training is the use of different media to present the information. The workbook serves as a central point for all the concepts presented during the course, regardless of the media. A well laid out workbook will also provide a relevant place for notes on the subject being discussed. When the class is over, the workbook should make a good reference resource. All the student's notes on a subject will be placed on the pages that reference the same topic in the course.

Advantages Unique to Public Education

There are positive aspects that are related to public classes alone. Interruptions are minimized, new ideas and situations are encountered, and peer level resources are established with the other students.

When the student attends a class at a public site, it is possible to become fully immersed in the subject matter. Interruptions, especially from co-workers who "must" have an answer, are kept to a minimum. This allows each student to concentrate on the material being covered, instead of problems at work.

Sharing the training with students from other companies offers some distinct benefits. People come to the class with different sets of experiences and needs. A point of view that is different from their own can be very beneficial. Some material presented in every course is difficult to apply to their own situation. This fact makes the retention of potentially useful material unlikely. Seeing the material from another student's angle may help to make it important enough to remember. Sharing different approaches to the same subject may also offer new productive ideas that are not directly related to the subject matter.

Each student is a potential new resource to every other student in the class. The best way to solve any problem is to approach it with the help of someone who has already encountered a similar problem. The contacts made during a public course could prove to be invaluable.

Advantages Unique to On-site and Custom Education

On-site education offers benefits that public does not. The greatest advantage is the ability to tailor the class to your needs. If you have paid for a class and your staff members are the only ones attending the course, you should have some say in what is being taught. It is possible to focus on specific issues that may only be relevant to your staff. With a little time spent prior to the class, you can make the instructor aware of anything specific that needs to be covered. You also have complete control over the whole learning environment: the course logistics, the classroom, the equipment used and the hours. Custom education actually amplifies the positive benefits of the on-site education. It is possible to tailor the course material to your specific needs and environment.

Disadvantages of Professional Classroom Education

The least favorable aspect of professional classroom education is that of control. You have little or no control over the material unless you take the initiative and are able to spend the money to have a course customized. In general, this type of education is "canned," developed by one person and taught by many. Therefore, the course quality is dependent on the trainer's understanding of the material and the developer's intentions. Adjustments must be made by the instructor as the course is conducted. To ensure a uniform presentation of the material from one class to the next, some vendors discourage the inclusion of too much material that is outside the original course's scope.

Disadvantages Unique to Public Education

In public classroom education the limited control extends to the instructor, facilities and other students.

While most companies that offer professional classroom education strive to employ only the best instructors, you have no control over the instructor that is used to teach a public class. If the instructor is less than satisfactory or does not present the material in a proper manner your only recourse is to request a refund. The same is true for the education facility.

While the presence of students from other companies can be seen as a positive factor, in some instances they could be detrimental. For example, if one or two students manage to dominate the class and turn all discussion to their own specific needs, the whole class suffers. Likewise, a single student's poor attitude can affect the entire class. Since each student has paid the same amount for the class, the instructor can only try to make sure each student receives equal time. While disruptive students are rare in professional education, it is even more rare for one to be asked to leave a class.

Disadvantages Unique to On-site Education

There are two major concerns associated with on-site education, interruptions and facilities. Both can greatly affect the quality of the course and the amount of material that is actually retained.

Of the two concerns, interruptions are the most serious. It is very difficult to totally shield students from interruption when the course is being held at their workplace. It may be possible to notify other personnel that your staff is unavailable during the class. However, it is impossible to keep them from cornering your staff during a break, lunch or after class. This can have the same effect as interrupting the class if your staff can not put daily problems "away" while they should be concentrating in class.

Many companies do not have the facilities to conduct a class properly. It is expensive to maintain a classroom and equipment unless it is in use the majority of the time. "Make do" facilities usually have a detrimental effect on the learning experience. For example, overheads that are displayed on the wall are hard to read; temporarily mounted whiteboards or chalk boards are difficult to write on. Likewise, crowded classrooms are almost impossible to maintain at a constant and comfortable temperature, make it difficult to take notes and difficult to hear and see the instructor. Each of these situations makes learning and retention that much harder.

Group dynamics can have a negative affect on learning in on-site training. When learning with a group of peers, who are potential rivals and maybe even the boss, many people are hesitant to ask questions. Further, some people perform poorly when the instructor tests for understanding if they feel that their peers or superiors are going to judge them. On the opposite end of the spectrum, some groups get along too well and it is difficult to keep everyone within the scope of the course. Thus, on-site group dynamics can negatively affect the amount of useful material that is retained.

PROFESSIONAL SELF-PACED TRAINING

There are three types of professional self-paced training available: workbook, computer based and video. Each has some characteristics in common as well as specific advantages and disadvantages. All three types of self-paced training can be customized by the vendors that provide them.

Workbook training is fairly self explanatory. Such training could be a purchased study guide that was developed for self-paced use. It could also consist of borrowing someone else's workbook from professional classroom training. It could even be as simple as going through a technical manual from front to back.

Computer based training is a tutorial program designed specifically to lead the student through a learning process. These tutorials can reside on a mainframe, mini or personal computer. They may or may not have anything to do with the computer system that they run on. They usually consist of carefully designed screens that use both verbiage and graphics to present the material and are interactive. Some computer based training is not interactive, it is simply text and graphics, displayed one screen at a time. This type of training should be considered an "electronic" workbook when considering advantages and disadvantages.

Video training can take on many different forms. It may be a tape of someone teaching a live class. However, many professionally developed self-paced video courses are developed specifically to take advantage of the technical capabilities of video.

Advantages of Professional Self-paced Training

There are advantages to professional self-paced training: it is developed professionally, down time is minimized, course availability is high and material presentation speed is self-paced. There are also some advantages that are specific to computer and video based training.

The advantages of professionally developed self-paced training are similar to any other course that has been developed by professionals. Course flow and approach, content reinforcement, iterative development, and course notebooks are only limited by the medium. Each learning technique would receive the same emphasis presented with professionally developed education.

A fairly large amount of down time must be scheduled for classroom education to take place. This is not true when self-paced training is used. The training can be broken into small enough segments so that it is easily scheduled. Most self-paced training can take place in the student's office, so they are never really out.

Once the self-paced training is purchased it can be used again and again. It is available for anyone to use. People can go through it several times if they find that they need a refresher course. New hires also can take immediate advantage of self-paced training.

Each concept in a self-paced course can be studied as long as is necessary for the student to achieve understanding. If the concepts are not clear at the end of a module, the entire module can be repeated. In contrast with classroom training, this can be done without the worry or concern that someone is going to judge such actions.

Advantages Unique to Computer Based Training

Computer based training has two specific advantages, it can be interactive and it can test online for understanding. Both of these are more difficult to achieve in either workbook or video based self-paced training.

Because the computer can evaluate the student's input, computer based training can be interactive. This is especially helpful when teaching syntax based material (e.g. computer languages and operating systems). The student can get a "hands on" education that is interspersed with helpful prompts and information. Some computer based training actually uses a repetitive approach - the action is "typed in" by the computer, then the student is asked to repeat it. It has been proven that adults learn best by doing. Interactive computer based training emphasizes this type of learning activity.

Interactive computer based training can also test for understanding. Because the computer can interpret the student's input, it is possible to respond to incorrect answers or actions. This type of programmed response is limited only by the developer's imagination. Programmed responses can take the form of gentle chiding, prompting, hints, restating the question, or simply displaying the correct answer. End of module tests can also be used to evaluate for understanding. Depending upon the score, the student may be advised to repeat the module or continue with the next module.

Advantages Unique to Video Training

The main benefit of using video based training is that it can approximate an important part of the classroom experience. One important principle of adult training is the use of different visual approaches. The combination of audio and visual examples is also important. Video is an excellent medium for combining the use of audio and multiple visual learning experiences.

Disadvantages of Self-paced Training

Some of the major weaknesses that are shared by all forms of self-paced training are a lack of: continuity, reinforcement, re-iteration and interaction. There is also the problem of ensuring the information contained in the training is current.

The greatest weakness of self-paced training is the lack of continuity. This is especially true if the training takes place in an individual's work area. It is difficult to refuse interruptions, especially if they are business oriented. Most people do not have enough discipline to close the door, stop incoming calls and really concentrate on self-paced training.

Some people try to address the problem of distractions by using the self-paced training after work. While there may be fewer distractions and interruptions, no one is really fresh and ready to learn after a hard day of work. Even if there are relatively few distractions, it can be hard to turn off other problems and really concentrate. This is especially true if the training has been scheduled for a short block of time.

An important principle of adult education is that new concepts should be reinforced. Reinforcement is usually achieved by testing for understanding or some other interaction with the instructor or other students. Neither an instructor nor other students are present with self-paced training.

The lack of true interaction is another shortcoming of self-paced education. Presenting complex principles or theories is especially difficult to do without some interaction. While computer based training does a good job of eliciting responses, it is not truly interactive. Without the stimulation that is provided by interaction, discussion and questions it is difficult to maintain good concentration for any length of time.

Professional self-paced training is always available once it has been purchased. However, most people only use it once. Because of these two facts it is easy to forget that it may need to be updated. If a maintenance or update plan is not included with professional self-paced training it is easy for the course to become dated without anyone knowing it. Most people think that the purchase cost of professional self-paced training is a "one time cost." This attitude also contributes to many companies using self-paced training long after it is outdated.

Disadvantages Unique to Workbook Training

Workbook study provides a fairly low retention rate due to several factors. It has been proven that adults need a change of presentation techniques at least every 20 minutes to maintain a high attention level. Of course, a change of techniques is not possible when working through a workbook. Using an old course workbook to learn from can be frustrating. Workbooks that are developed to accompany classroom training often contain only bullet items and major points. It is assumed that the instructor will provide the detail information during the presentation.

Disadvantages Unique to Computer Based Training

Computer based training takes more development hours for each training hour than any other form of training. A conservative estimate is a 125:1 (yes, that is 125 hours of development for each hour of training time). Because of this large investment most computer based training is not kept current. Therefore, it is important to make sure that the computer based training is written for the same version of software or hardware that you are using.

Disadvantages Unique to Video Based Training

While video is a wonderful medium for many good visual training techniques, it is also the easiest to tune out. When the tape is running, no action from the student is required to "complete" the course. The fact that most videos are shown in a darkened room makes concentration even harder and less likely. Most people are overextended and a darkened room only serves as an invitation to "rest your eyes."

TRAINING DEVELOPED IN-HOUSE

Education that is developed in-house is usually in one of three formats: classroom, self-paced or ad hoc. Classroom and self-paced training that are developed in-house have the characteristics of public training previously described.

Ad hoc training could also be considered on-the-job training. It is any informal training that takes place during the normal business day. This type of training is important and can be very effective in specific circumstances. Ad hoc training is not really under the manager's direct control. It is also impossible to plan for and does not affect the training budget. Therefore, it will not be considered in this paper.

Advantages of Training that is Developed In-house

There are some very positive aspects that are associated with training that is developed in-house. The training is customized, and your company has full control over the content, instructor and facilities.

A very good reason to develop training in-house is that it can be customized. The training can focus only on the areas that are relevant to your company. Special techniques, ideologies, policies and procedures can be emphasized. Proprietary information can be included with generic materials. Your staff can be introduced to "real life" uses of different products through company examples.

Control over the instructor and facilities is another strength of education that is developed in-house. Because the course is being taught by your employees at your facilities you can ensure that the training is being delivered properly.

The positive aspects that were presented for both professional classroom and self-paced training apply to education that is developed in-house with one exception. Most companies find it too costly to employ a professional classroom developer. Therefore, the positive aspects of professionally developed courseware can not be assumed when training is developed in-house.

Disadvantages of Training that is Developed In-house

The major drawback to developing training in-house is the time and expense that is associated with developing and delivering training. Recall that a good course is dependent on professional development and delivery. The fact that it is quite expensive to develop a professional quality course leads some managers to try to cut corners. This is the real danger in developing training in-house.

Many managers want to use one of their staff members who is proficient in a certain area to train others. The manager is unwilling to dedicate the time it would take to produce a good course. The staff member probably does not possess the skills that are needed to develop or deliver the training. Training that is done with little or no preparation is worthless.

Disadvantages of In-house Development

Course ware must be developed in a professional manner, or the impact of the training can be greatly decreased. This is not to say that all courseware developed in-house will be inferior to that developed by professionals. However, you can not assume that all the principles that are used by the professionals will be known or considered by someone who only knows the subject matter.

An understanding of the subject matter is only one part of developing a successful course. There are many principles of adult learning that must be used to ensure a good learning experience. The timing of the course is also important. These are all things that can be learned in time. However, the courseware produced in the interim it will suffer. It would be better to take advantage of the companies that are in the business of teaching others to successfully develop training.

The ability to develop quality courseware is only one part of the formula for a good training course. Another important component of course development is time. Classroom training requires at least a 50:1 development ratio. The ratio for developing computer based training is 125:1. Video training requires a ratio somewhere between classroom and computer based training, depending on the complexity of the course and the video effects being used. At the very least a video course must be developed, then recorded.

Assuming the 50:1 ratio to develop classroom training, a two day course would require over 5 months to develop. Most companies would not be able to dedicate one employee for this amount of time. It is also difficult to anticipate the need for a course far enough in the future. Remember that the 50:1 ratio is a minimum, the development of technical courses could easily take longer.

Keeping the courseware up to date also becomes a factor. The time needed to update is not the only concern, you also must be sure that the developer understands the new material well enough to add it to the course ware. This would require some self study at the least, perhaps additional professional training for the developer.

When either video or computer based training are considered, in-house development becomes even more difficult. Few businesses are equipped to properly produce a professional quality video. There is quite a bit more to video production than owning a video camera. Computer based training takes special skills to develop. This is especially true if you want to use specialized graphics or other interfaces.

Disadvantages of Delivering In-house Training

There are two areas that can affect the quality of training when it is delivered in-house. The instructor may not employ good teaching techniques and the facilities may not be conducive to learning.

The old adage that "just because you are knowledgeable does not necessarily mean that you can teach someone else" is unfortunately true. Some people are natural teachers, others have a hard time sharing their knowledge with others. Regardless, if someone is going to provide training, it is imperative that he or she receives some "train the trainer" education. There are many principles that should be applied when training adults. Not all of these principles are intuitive. Without the proper training, an instructor can be at least ineffective, at worst dangerous. Good retention can not be expected if the students are frustrated with the instructor. For example have you ever taken a college class with an instructor who does express himself well?

The training facilities can also be unsuitable. As mentioned earlier (in On-site Education) a classroom that is improperly sized or set up in a makeshift fashion can be quite a detriment to the learning process. It is important to consult some authorities and do some long range planning before designing your facilities. For example, most people would set up a "classic" classroom (with the desks or tables in rows) by default. Believe it or not, this is the worst possible configuration when training adults.

COMPARISON OF TRAINING COSTS

The best way to look at the costs involved with the different types of education is in a graphic form. The following graphs display the relative cost per student for professional classroom training (both public and on-site), professional self-paced training (workbook, computer based and video), and in-house developed training (classroom, computer based, and video). A different graph will present the data for each of five, three and one training days.

The following costs were used to develop the data represented in the graphs:

Purchased Classroom		
	- public	$250 / student day
	- travel	$500 + 100 / day for each student
	- on-site	$125 / student day
Purchased Self-paced		
	- workbook	$50 / student
	- computer	$1000 / day
	- video	$200 / day

In-house Development

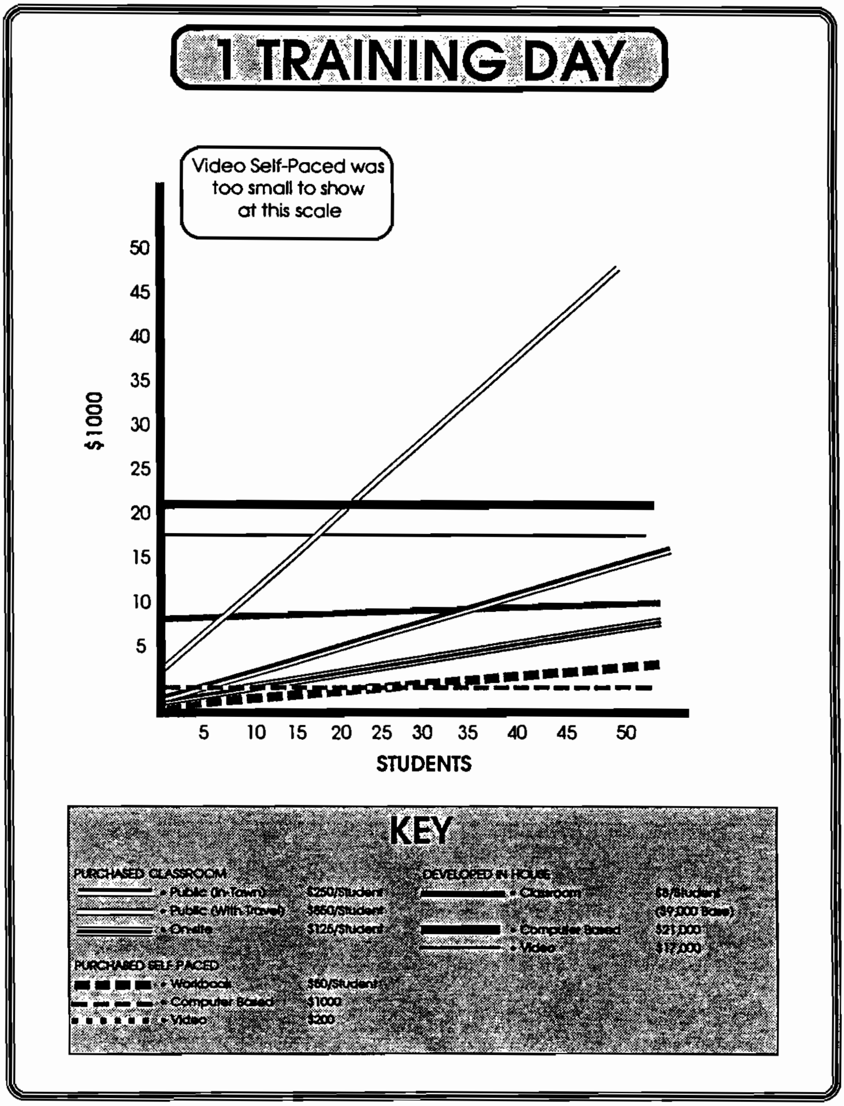
- development hours (development : training)
  - classroom 50:1
  - computer 125:1
  - video 100:1

- developer cost $40,000 / year (240 working days)

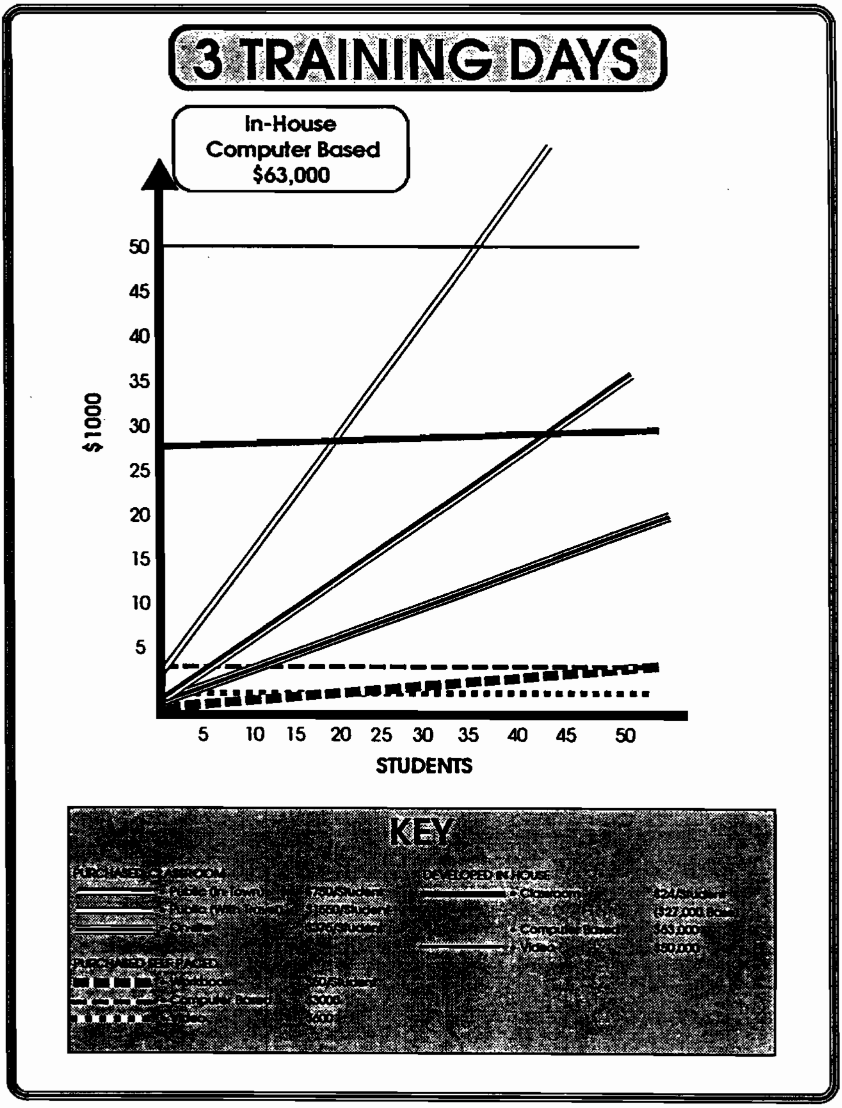
- typesetting/format $4 / page if done in-house
  ($20 / page for professional work)

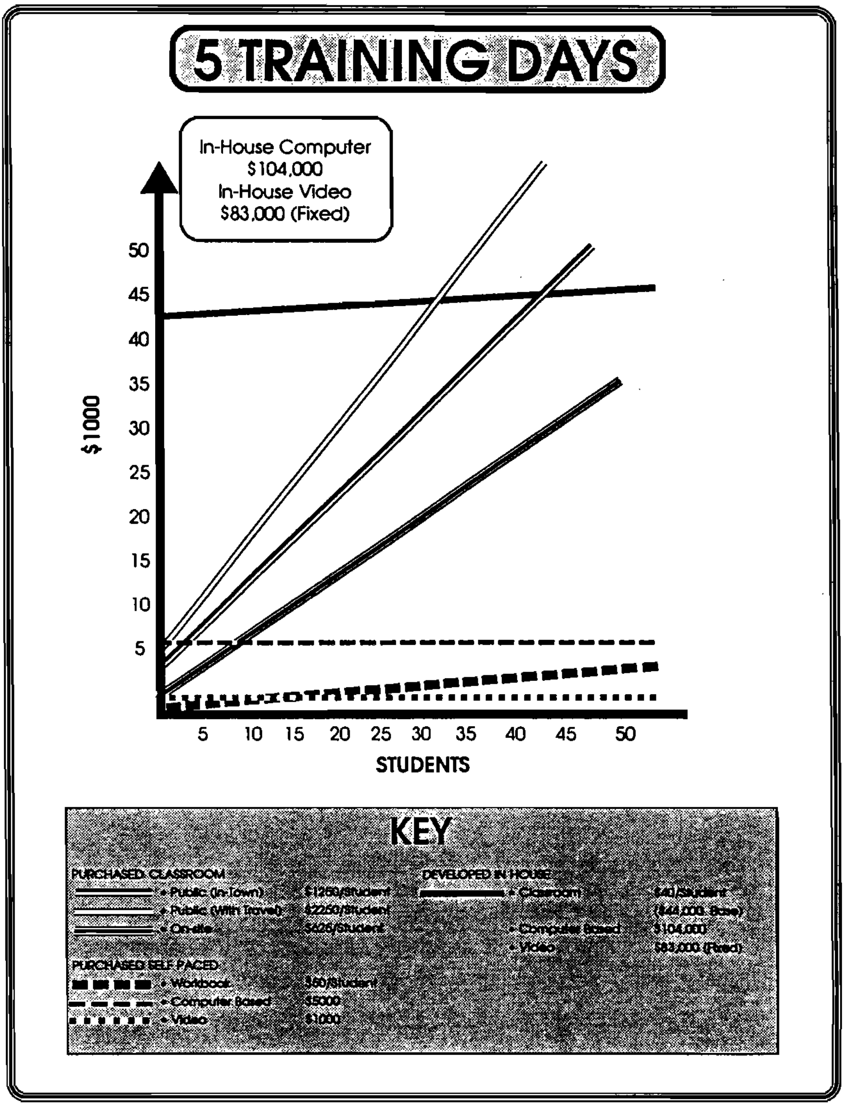
- materials cost .05 / page
  160 pages / class day

The cost of purchasing the proper equipment to produce video self-paced training in-house has not been included. It is assumed that this would not be attempted if the company did not already own the equipment.



1 TRAINING DAY
Video Self-Paced was too small to show at this scale
$1000
50
45
40
35
30
25
20
15
10
5
5
10
15
20
25
30
35
40
45
50
STUDENTS
KEY
PURCHASED CLASSROOM
• Public (In-Town) $250/Student
• Public (With Travel) $850/Student
• On-site $125/Student
PURCHASED SELF-PACED
• Workbook $50/Student
• Computer Based $1000
• Video $200
DEVELOPED IN HOUSE
• Classroom $8/Student ($9,000 Base)
• Computer Based $21,000
• Video $17,000


3 TRAINING DAYS
In-House
Computer Based
$63,000
$1000
50
45
40
35
30
25
20
15
10
5
5
10
15
20
25
30
35
40
45
50
STUDENTS
KEY
DEVELOPED IN HOUSE
Classroom
Computer Based
Video



Education: In-house, Out-house, or CBT?

These graphs do not take time into consideration. This adds another dimension that is difficult to show graphically. For example, if the break even point for five days of classroom training (developed in-house versus public classroom training) is 22 students, it is important to consider how long it would take for those students to need and take the course.

If you put 22 students through the same class in one year, the graph is fairly accurate. If it will take five years to put 22 students through the class, you must add in the cost of updating the class at least two times. This cost would change your break even point.

Another important time related factor is development time. Can you afford to wait until the courseware is developed? Development time for a five day class is a little over one year.

CONCLUSIONS: WHICH ONE IS BEST?

As with most of the decisions that managers must make, there is no clear cut best choice when planning for education. It is necessary to take into account what type of education you need, how many students you want to train and how fast you need to get them trained.

There is one type of education that is best when you look at retention rates alone. Most experts agree that classroom training which includes hands-on workshops and interactive case studies has the best retention rate. Either purchased classroom training or that developed in-house is acceptable, as long as the quality is equal. Classroom training is the only type that is able to incorporate all of the important principles for successful adult learning. Some of the more important principles include: setting clear objectives, showing the relevance of the information to the student, interactive responses, frequent changes of method or media, use of as many senses as possible and 60% - 70% of the training time devoted to application or feedback.

The experts agree classroom training is the best. The graphs show that the break-even point for classroom training that is developed in-house is over 20 students. Does this mean that purchased classroom training is the only answer for small to medium sized companies? **NO, THIS IS NOT NECESSARILY TRUE.** It is important to look at what you need from the training before you make a decision.

Professional Classroom Training

Professional classroom education is the best choice only if the training you need is product or equipment oriented, not company specific. It is important to remember that most purchased classroom training is designed for general audiences. It is most beneficial to have a special follow-up course that is developed **in-house** to help people apply what they have learned in a public class to your situation.

Customized classes are also a very good alternative. This can give purchased training a "home grown" feel. Some training courses can be purchased and taught by your staff. This allows even more freedom to add information that is specific to your situation.

Purchased Self-paced Training

Purchased self-paced training is such a good buy that it is hard to resist. It is best when it is used for short (less than 1/2 day) training sessions. This makes self-paced training an excellent means to give an overview of any subject. It can also be used to introduce new material. Many people use self-paced training prior to attending purchased classroom training. This allows them to get the most information possible out of the more expensive training.

Self-paced training is also a good tool when used to refresh knowledge. Concepts that have been learned previously but have not been used can be reviewed quickly in a self-paced format. It is also possible to present short clearly defined subjects with self-paced training. New concepts contained in a version update would be a good candidate for self-paced training.

Education Developed In-house

The best use for education that is developed in-house is teaching proprietary information. This could include policies and procedures, data structures, and specific coding or file naming conventions. These are all things that are specific to your situation.

The expense involved in developing training makes trying to produce courses that are already available in the marketplace unfeasible. If you want the control that internal education offers, it makes sense to purchase courseware and customize it or have it customized when ever possible. Remember the old adage, "There is no sense in trying to re-invent the wheel."

TITLE: Sales Force Automation Today

AUTHOR: Michael Barry

Gateway Systems Corp.

2400 Science Parkway

Okemos, MI 48864

517-349-7740

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 5195

EDI in the Real World

Wayne M. Swafford
Larry W. King

Computer Management Solutions
7208 S. Tucson Way Suite 175
Englewood, CO 80112
(303) 790-0123

Introduction

Most people have heard about EDI and know what EDI is. But what impact will EDI have on a company? What will happen when EDI comes in? How should I implement it? This paper will try and address those issues and more. For those who know very little about EDI, the first section will discuss the basic terms and ways to complete the transfer of data. The second section will look at the effect of EDI on different departments in a company, and the third section will talk about the implementation options available to make EDI operational.

EDI - What is it?

EDI stands for Electronic Data Interchange (or Electronic Business Data Interchange). EDI occurs whenever a company exchanges its own computer-stored information with that of some other company. Information may be exchanged between a business and its suppliers, its customers or internally. While in its simplest form the actual magnetic storage media (tape, cartridge, disk, floppy diskette, etc.) may be physically transported, today EDI more often refers to the real-time transmission and reception of data between computer systems using data communications equipment (modems, multiplexers, phone/data/satellite communications lines, etc.).

Although EDI is a technical subject, it's important to understand that it is primarily a powerful business strategy. When properly implemented, company information is more timely and accurate, and can result in reduced expenses, increased sales, enhanced strategic and tactical marketing and better-informed overall decision-making. In short, effective EDI improves the bottom line.

Terms

Understanding EDI requires some explanation of basic terms. Here are a few along with brief definitions.

ANSI - The American National Standards Institute, Inc. This organization pools resources from companies around the country to establish standards on a wide variety of issues. The Accredited Standards Committee X12 was charged with the development of transactions formats for use in an Electronic Business Data Interchange (EBDI) environment. ANSI has also established the Data Interchange Standards Association (DISA) to work with the X12 committee in creating EDI standards.

Data Element - The most basic unit of information in an EDI environment. Multiple data elements comprise a data segment.

Data Segment - The intermediate unit of information in a transaction set. A data segment or record is comprised of logically related data elements or fields.

Dictionary - A data structure which stores the parameters necessary to permit data communications with another host computer.

Direct-connect - An EDI method where two computers communicate directly with each other, generally using modems and voice-grade telephone lines.

Group - A collection of like transactions to be electronically interchanged. Examples of groups (or group sets) are invoices, purchase orders, sales orders, shipping manifests, quotes, etc.

Private Data Format - Any non-standard EDI record layout established for the sole use of a company and its trading partner(s).

Session - The period during which a company's computer is linked to another processor or network for the purpose of EDI.

"Sneaker Net" - An affectionate name for the primitive form of EDI where tapes, cartridges, diskettes, etc. are generated by a company and physically transported (via courier, mail, or hand-delivery) to its trading partner for use on their system.

Trading Partners - The two or more companies which engage in Electronic Data Interchange.

Transactions - Also known as transaction sets, one or more transactions comprise a Group. Examples of transactions are individual invoices, purchase orders, quotes, etc.

Value Added Network - Often called VANs, Value Added Networks are third-party services which provide a safe, reliable environment for EDI transmissions. VANs often employ a 'mailbox' type of arrangement where both trading partners have an electronic mailbox on the network, which can be polled periodically for EDI information. Some of the larger VANs include Geisco (General Electric Information Services Company), McDonnell-Douglas, Tymnet, Telenet and Compuserve.

Types

EDI can be accomplished in three basic ways using two fundamental format types. The three ways are Physical Media Exchange ("Sneaker Net"), Direct-connect Data Exchange and Value Added Network Data Exchange. When EDI is executed by either of the latter two methods, the interchanged data will either comply with some standardized format (e.g. ANSI X12, EDIFACT, TDCC, etc.), or it will utilize some private data format established by one or more of the trading partners involved. Let's examine each method and format briefly.

Physical Media Exchange involves the creation of some form of magnetic media on one trading partner's system. The media may be a magnetic tape, cartridge or digital data storage cassette, a hard disk pack or a floppy diskette. This media is then physically transported to the other trading partner's site, where the media can be loaded onto their computer and processed. Once processed, the receiving partner can then create updated media to be transported back to the original trading partner. While this method is somewhat primitive, it can be quite effective in eliminating redundant data entry and clerical error. Since 70% of computer input comes from computer-generated documents, this advantage alone can be significant. However, some substantial risks are also apparent with this method. The physical media can be mishandled and not reach the trading partner's site in a timely fashion. Also, the media is subject to moisture, heat, dropping and other mishaps which can destroy data. Further, even when this method is executed successfully, there is always an inherent time delay caused by having to transport the physical media from place to place. Typically, private data formats are used in this method, since only two trading partners are involved.

Direct-connect Data Exchange most often describes a situation where a trading partner's computer system uses a modem and telephone lines to directly dial-up the system of another trading partner. While the two computer systems are linked, data is transmitted in real-time between systems. Sometimes this may be as simple as building a batch file on the receiving system to be processed later. More often, however, the two computers will engage in some kind of interactive dialogue. That is, for each record or data segment transmitted by the sending computer, the receiving computer sends back some response. In some cases one trading partner may dial-up and run the interactive systems of another trading partner. This generally requires some sophisticated software to allow the sending computer to model the user environment of the receiving system. Private data formats are most often used in this method as well. A disadvantage of the direct-connect method is the fact that the trading partners' systems may be geographically separated which can result in huge long-distance phone charges. Also the communication lines between trading partners are rarely error-correcting or conditioned, which means that data transmissions are subject to line noise, power fluctuations, etc. This can lead to unsuccessful transmission and/or data corruption.

Value Added Network Data Exchange is probably the most commonly used EDI method today. In this arrangement, The sending trading partner will dial-up (often through a local phone number) a value added network, which acts as a clearinghouse for EDI transmissions. The data is transmitted into a "mailbox" where it waits to be polled by the receiving trading partner. The receiving partner's polling process reads the mailbox and transfers its contents to its own system for processing. Once processed, the updated information is then transmitted back by the receiving partner to the sender's mailbox on the VAN. The original sending partner connects to the network and polls his own mailbox for the updated data. This information is used to update the sending partner's system. This method is the most commonly used for good reason. The network provides a way of exchanging data with multiple trading partners without tying up ports for each one on the sending system. It also generally involves local phone numbers, which eliminates costly long-distance charges, and it often utilizes conditioned lines and error-correcting modems which help to insure effective transmission. Standard and private data formats are used. However, standardized EDI formats are encouraged here because the number of potential trading partners is virtually limitless. One of the advantages of a standard format is that data can be exchanged quickly and easily without reformatting for each trading partner.

Business Effects

The business effects of EDI are many and varied. If the business is not ready, the results can be disastrous. The major business areas effected are, the Data Processing department, the Accounting department, the Marketing (Sales) department, and the department(s) whose function is being automated or replaced by EDI.

Data Processing

The Data Processing department has two different areas that can be effected. Operations and Applications. Some smaller DP shops only have an operations area, that area will be looked at first.

Operations

In many cases, Hewlett-Packard computers are considered 'operatorless'. As operatorless machines, many operators don't pay attention to what happens on the machine. My first job on a Hewlett-Packard computer was as an operator. I know that as an operator, I hardly ever paid attention to the console because hardly any important information was ever sent to the console. The operators must be retrained to pay attention to the information that may be sent to the console.

New Complexity

Adding EDI to a computer adds a level of complexity that does not exist on a non-EDI machine. Data communications is not an exact science. When a system deals with modems, other systems, more modems, still other systems, there are bound to be occasional problems. When an EDI system talks to a standard network, no less than three systems and three sets of modems are involved. At some point in time, one of those links will fail and that is the time when the operations department needs to be alert and ready to act. Not only does EDI deal with other systems, but no matter what method of EDI transmission is used, it still communicates through the phone lines. When dealing with the phone company, all kinds of problems from storms to sunspots can play havoc with the transmissions. All of these problems need to be monitored by the operations department. When a business is highly dependent on its EDI applications a down system can have similarly critical effects as a down phone system.

Processing Load

Adding EDI to your system will add additional load to the CPU. This load manifests itself in two different ways: the load of the EDI communications, and the load of the applications themselves. The method of performing EDI determines the amount of load placed on the system. Some methods are very CPU intensive while others are much less so. When the decision is made regarding which method to use, the CPU load should be considered. The different ways of performing EDI will be discussed later on in this paper.

Applications

There are three different ways for EDI to be installed. The programming could be done in-house, an off-the-shelf package could be purchased, or a custom software package could be written by a third-party. The first step in this decision is to determine the type of EDI supported by the trading partner desired. Then, after that is determined, the option needs to be chosen.

In-House Development

If the choice is made to develop the software in-house, the expertise level of the programmers in a very real concern. The applications programming is not that difficult, but the communications software is very difficult. If all of the programming will be done in-house, the programmers need to know more than just how to write standard HP Cobol. The communications software is very in-depth, close to the machine code. A long, hard look needs to be taken before the decision is made to write all of the software in-house. The complexity of data communications and the constantly changing EDI standards can make this an extremely difficult and expensive proposition. If the interface is built in-house, the programmers also have to be able to respond quickly to changes made by the trading partner or by the communications vendor.

Standard Package

If a standard package is chosen, the changes required to the existing software must be looked at. If the software in use is from a third party, the implementation of EDI must be made within the constraints of that third-party software. But, if the software is developed in-house, the possibilities are limitless as to the additions and changes that can be made to the software at the same time. Again, the decision must be made as to who will create the software to interface the package to the standard software used.

Third-Party Custom Development

If the decision is made to have a third-party custom build and design an EDI package, a few things need to be considered.

* Do you want the third-party to write the interface into the standard software or just the communications portion?

* Can the third-party vendor respond quickly to changes made necessary when the trading partner changes things?

* If necessary, can (or will) the vendor build the custom interfaces to the existing software?

* What kind of a track record does the vendor have in the EDI community?

* Who will be responsible for on-going support of the software?

Data Processing Summary

The effects on the data processing department when EDI is implemented are many and varied. The major day-to-day effects are seen in the Operations area where the operators must be much more aware of the things running on the system and must be able to react quickly when a problem occurs. In the Applications area, the major decisions that need to be made is who will create the software. EDI opens up a whole new level of changes that can be made to the system. The software can either be created in-house, be a standard product, or a custom package, or it could be a combination of these three.

Accounting Department

Depending on the area of the company to be automated, major financial impacts can be seen in a company. Cash flow can be altered, merchandise can be brought in at the lowest possible cost, the error rate of transactions can be lowered. EDI can have all of these impacts. The accounting area will not be discussed too much except for the cash flow differences.

Cash Flow

Cash flow can be dramatically effected depending on which area of the company is automated. If the sending of checks to vendors is automated, the float on the dollars paid is drastically reduced, on the other hand, if receivables for large customers is automated, the time it takes to receive payment can be reduced. When determining what EDI transactions to automate, make sure the financial impact is considered, not just the cost of the system, but also the residual effects of speeding up these transactions.

Marketing Department

The marketing department is effected by EDI not in any real, concrete terms. But, there is an effect. Many contracts now require that EDI be used to allow the customer to transmit purchase orders to the supplier and to have the invoices transmitted back to them. Having EDI available will open up some of these companies to a business relationship. The marketing department can also use EDI as a sales tool to other customers. For example, with EDI a company can know exactly what items can be received from a supplier much sooner than by conventional methods. In one company, EDI has cut the number of packing slips sent to customers by about 20% by converting to EDI. This company also cut delivery time by 50%.

Automated Department

The department to be automated also has a major impact. Tasks now done manually are now completed automatically, less rework is done because the data sent is more accurate. Some companies have seen a major reduction in manpower. EDI can replace people, but if you want the new system to be received well, eliminating people is not the way to go. Reassigning them maybe, but not eliminating them. In one EDI site that we installed, a purchasing person was let go a week after the automatic ordering software was implemented. This person was not let go because of EDI, but it was assumed that she was. To this day, over two years later, the purchasing department still has not embraced the software because of this one incident. There is so much that this software will do for them, make their jobs less tedious, make them more efficient, but because they fear for their jobs, they won't use the power of the software. Making people more efficient, not obsolete, is the goal of EDI.

Other Departments

When a department of a company is automated, other departments are effected. for example, when the purchasing department is automated, the warehouse can be effected. In a system we put together for an office products company, their system was changed to hold all orders that had a backorder on them that could be received that same day from a wholesaler. The order would then print after the shipment was received from the wholesaler that afternoon or evening looking as if the item ordered by the customer was in-stock the entire time. This required a major adjustment in the warehouse. Orders that used to print in the morning now print in the late afternoon or evening. Adjustment had to be made both in the warehouse and in the software. Other departments were also effected. Accounts Payable now had correct invoices to pay from because the software updated the purchase order with the actual cost of the item and the amount acknowledged by the wholesaler. That was very positive. The downside for payables was that more invoices were received. The point is, when exploring the effects of installing EDI, look beyond the obvious changes necessary in the main department to the downstream effects. Sometimes those are even greater than the primary department.

Business Effects Summary

In summary, the business effects when implementing EDI can be both positive and negative. If the transition is planned well and executed well, the effects will be mostly positive, but if the planning and execution is poor, Watch Out! Problems can and will occur. As with any project, the more design work that is done on the front end, the better the project will be on the back end.

How to Implement EDI

Okay, now you know some of the gotcha's, how do you implement EDI. There are several different ways to perform the communications between the HP3000 and other systems. There are four common ways. Two require hardware other than an modem, two just software. The ones that require hardware are using an INP from HP and using a Network Engine from Telamon. The software only solutions can use Privileged Mode capabilities or just standard HP Intrinsics.

Hardware Solutions

The hardware solutions require an investment in hardware but could end up costing less on the software side. The two main ways of communicating to other systems using additional hardware are with RJE through an INP or by installing a Network Engine from Telamon.

INP - RJE, X.25, IMF

One of the most common ways to have an HP3000 communicate with another computer is with RJE or Remote Job Entry. RJE is a standard HP product that communicates with the outside world through an INP or Intelligent Network Processor. RJE has been described as the "only thing that really makes sense right now" for asynchronous communications by Frederick Ochs, Jr. in a recent issue of Interex magazine. I disagree. Yes, RJE is a viable choice, but it has its' limitations. Other choices from HP include, X.25 communications, IMF, or Interactive Mainframe Facility, and, I hate to say it, MTS. If a company would like to have more than one link to a trading partner active at the same time, each link would require its' own INP. That to me is a problem. RJE is a viable solution for a low volume of transactions, and only a few different trading partners, but it is not the best way.

Network Engine from Telamon

Telamon is a company that advertises that they can link anything and everything to the HP3000. In the dealings I have had with them, I must agree. They have two products that would work well for EDI, the Asynchronous Network Engine and the Synchronous Network Engine. These boxes have the same functionality of RJE but at a fraction of the cost. I personally have not used these boxes, but have talked with Telamon and have dealt with them in the past. From what I could tell, if a hardware rather than a software solution is called for, the solution from Telamon would be the one to choose.

Software Solutions

The EDI solution that I am most familiar with is the software solution. There are two primary ways to install EDI using software alone, except for a modem of course. Those ways are with Privileged Mode capabilities and with standard supported HP intrinsics. The problem with software solutions is that the programs must be written in such a way as to not miss data being transmitted from the trading partner since a port on the HP does not buffer data.

Privileged Mode

Programs written using PM capability solve this problem by performing the read before telling the remote system the receiving system is reading for data. This is called NOWAIT I/O. Another function of many of these processes is that they must use the Linear or 'B' queue of the HP to receive the performance that is necessary. This can cause severe system degradation, especially on heavily loaded machines. If the machine is heavily loaded, this may not be the best solution. Also, since this solution requires PM capability, it is not guaranteed to work with future releases of MPE. As such, it is a somewhat risky solution.

Standard HP Intrinsics

The final solution that will be discussed is the Standard HP Intrinsic solution. I can't go into much detail due to the fact that the information is proprietary, but I will try to give as much information as possible. Basically, this solution uses intrinsics in place on the HP without using PM capability. The data is captured without losing data, no hardware, except for a modem is required. A modem port isn't even necessary! One of the locations using this technology has had as many as seven links to different machines active at any one time. To do that with a hardware solution would be extremely expensive, with the PM solution, the system overhead would be unacceptable.

How to Implement Summary

In summary, there are four basic ways to do EDI, two hardware solutions and two software solutions. Each have their own strengths and weaknesses. The correct solution depends as much on the environment as the requirements of the trading partner. As the EDI solution is designed, be aware of the number of possible links and the impact of the machine. Ask the third-party vendor what type of technology is used in the product. Be aware that not all solutions are alike.

Conclusion

In conclusion, it is hoped that through this paper, EDI has become more real. It is currently a major force in the business market of which many companies have not yet taken hold. It can be used to leverage business, improve cash flow, or improve efficiency. EDI is not a toy. It is viable today and will be necessary tomorrow.

"Don't Be Cruel to a Heart That's True"

Diane Amos, C.P.C.
Amos & Associates, Inc.
633-B Chapel Hill Road
Burlington, North Carolina 27215
United States of America
(919)-222-0231

As a manager, do you have employees whose "hearts are true"? How do you show them that you appreciate them? What do you do to retain those valuable employees?

In today's HEWLETT PACKARD marketplace, one of the costliest factors that a company has to deal with is staff turnover. The dollars start adding up when you look at the cost of recruiting a new employee, training them and then assessing lost productivity. The days of remaining in a job for twenty years to receive "the gold watch" are no longer reality, and with our mobile society being what it is, the 90's may show this to be an even larger problem. The average tenure of DP professionals is now 2 1/2 years. How can you, the manager combat this epidemic? When your good employee is resigning, should you offer him a "counter offer" to stay? NO! This is usually a drastic mistake for everyone concerned. By putting yourself in this position you are giving your employee the message that he's only valuable when threatening to leave. Let's take a look instead at preventive measures that will keep that employee from leaving in the first place. The best way to do this is to examine the numerous reasons that employees give for leaving companies.

The #1 reason for leaving a job is boredom and lack of challenge. Do you have your prize employee stuck in a corner because he does his job so well that you don't want to rock the boat? Well, the boat may sink with that attitude. People need to grow and expand in their jobs to be truly happy, and keeping a valuable employee in a "rut" will surely cause him to look elsewhere for challenges. It takes some insight and planning to keep your employees challenged, but the investment is well worth the time and effort. Think of special projects that you can assign, new and innovative ways to improve your shop. Cross training of jobs and rotation of duties keeps things interesting. In addition, many successful managers tell me that the more they challenge their employees, the more they are challenged themselves by their employees. Are you afraid to allow your employees to stretch themselves thinking that they'll outpace you? What may really happen is that you'll end up stretching yourself!

The #2 reason is lack of appreciation. When was the last time you really let your employee know how much his work, his effort and his attitude was appreciated? Do you wait for the yearly review or do you constantly reaffirm him with large doses of sincere praise when he's handed you a job well done. Management studies have shown that daily support, encouragement and praise for even the little things keeps an employee on an even keel. It's often the little things that count, and there are several small things that you can do for employees that may make a difference. Some ideas may be to take your employee out to lunch, or give him the day off. If your city has a baseball team, you might give him baseball tickets in his next paycheck. It's not the dollar amount but the thought that counts here.

Lack of recognition is a very similar reason. When an employee goes the extra mile to finish a difficult project or solves a problem that no one else has been able to solve, he looks for some recognition that his effort was noticed and appreciated. Again, you do not have to go to great expense to let an employee know he is valued in the company, as little things that are said or done often mean more than you think. A special parking spot for the "employee of the month" costs you nothing, but makes that employee stand out. Also, choosing the employee who is doing the steady, productive job for you to go to that next LUG, RUG or INTEREX meeting rather than the same ones who are usually chosen, may give that employee the feeling of importance that is needed. Recognizing an employee in a staff meeting of his peers with a plaque for "special service" costs next to nothing. A word of caution here: it is often common practice to "take turns" in recognizing employees so that everyone gets covered. This is a mistake! It diminishes the impact and the value of the reward. Recognition is meaningless when it's given for mediocre performance and when it becomes commonplace. A very productive way to recognize an exceptional employee is to let them prepare a presentation for a user group meeting. Start with the presentation within the department. This action will recognize the employee and also train the other staff members thereby killing two birds with one stone. A success story that I can relate is one of a rather quiet, introverted System Manager who once he warmed up to you was a brilliant, funny and witty fellow. His manager put him in the position of heading up a panel at a User Group. What a pleasure it was to see this employee blossom under this new position and swell with a feeling of importance. Being creative with ways to "recognize" your valuable employees now may save you the need of looking for creative ways to replace them later.

Another reason is a lack of input in the decision making process. If Joe has a job to do but no input on the best way to do it, his frustration level will grow. People want to participate and have a voice in the decisions that affect them. Of course, that isn't always possible, but spreading the decision making power to the lower levels will instill a sense of ownership, responsibility and accountability. As Nancy Austin, co-author of "A Passion for Excellence" discussed at the Interex Management Symposium: the companies that will grow and thrive in the 90's are those that will share the decision making power with their employees. "Trust" is a key issue here. Trust your employee to handle privileged information. You can't ask people to solve big problems without letting them solve little ones. Ms. Austin indicated that the companies who will have the **edge** will be those who nurture individual contribution, creativity and quick decision making. Critique your own management philosophy. Are you doing this?

Lack of direction from management is often heard as a reason to leave. While you're looking at your management style, check to see if you are being clear with your expectations and consistent in your demands. Monitoring your employee's progress on a monthly basis rather than at yearly review time will help to keep this in check and avoid room for misunderstandings. Have a two-way discussion to determine that you both are on the same wave length in how and when you intend to achieve your goals. Communication is the key, and having predetermined goals and objectives that you both agree on will avoid this problem. A performance plan should be in place and it's extremely important that the performance plan is a formally written document detailing the performance standards that the employee is expected to meet in order to achieve his goal. This then becomes the basis for performance appraisals which you can measure against. Employees love them, because if their boss dislikes them, they have something to put their hands around. It forces the boss to treat all his employees equitably and measure them by their performance rather than by emotion.

Matching your employees working style to your management style may help to avoid problems in the future. First of all, determine your management style. Managers should be both people oriented and production oriented. Look at the diagram:

PEOPLE		
	DEMOCRATIC	PARTICIPATORY
	POLITICAL	
	BUREAUCRATIC	AUTOCRATIC
	PRODUCTION	

Is your style Democratic where you have more concern for your people than for production? Is it Autocratic where you care more for production than for your people? Or is it Participatory where you're concerned for both? Or Bureaucratic where you don't care about either your people or production? Or is it Political where you flip-flop between all four styles at any given time? Having your applicant describe his ideal boss may help you to make that right hire and match the style at the very beginning. This may save you endless struggles of philosophical differences that no one can solve.

No career path? I hear this reason for leaving quite often! This is a difficult area in the HP market because there are so many small shops with few vertical career paths to offer. People tend to think of career paths as straight up when sometimes they're sideways and at angles. Discuss openly and freely with your employee what he truly wishes to do with his life and career. Perhaps you'll find that he could progress into the user area and would be happy to do so. Then put a career plan in place where he can take concrete steps to prepare for this next step. Don't make empty promises, but give a clear direction on how he can reach that goal. Make it a formally written plan detailing what your employee wants to do, and what he must learn or achieve to get there. Does he need to take a certain course or learn a particular skill? Where will he be able to go if he masters that? Be very specific and clear. What if your valuable employee wants to progress, but stay in a technical field and not go into management? A long term employee brings added value to the company and he shouldn't have to change jobs to have a career path. Raising the upper limits on salary and increasing the responsibilities and decision making input for people in the technical field could keep these employees satisfied.

A poor working environment is often given as a reason to leave. As manager, take a hard, critical look at the workplace that you have to offer. The best way to do this is to actually work in it for awhile. Is it conducive to good work, and a place where people will enjoy coming to work, or does it leave something to be desired? I've heard horror stories of computer rooms with desks crammed into them making the employees feel like they were in jail cells; dark and dingy cubicles stuck off in a corner somewhere far from where they needed to be to be effective with users; and I've seen offices that should house 1 programmer have 4 people stuffed into them. Issues that should be addressed and looked at carefully are the temperature - (too hot or too cold?), noise level- (too noisy?), color - (drab and lifeless, or offensively unnerving?), smoke-filled?, music - (irritating or conducive to efficiency?), traffic - (people tripping over each other?) and the general esthetic nature of the place. These are often more subtle reasons for leaving but they work on the frustration level of an employee. Have you ever been bitten by an elephant? No, it's the gnats that get you, not the elephants.

Are you giving your employees valuable training and sending them to the necessary schools, or are you expecting them to pick up everything on their own? It's been said that people are most happy doing what they know how to do well. If they feel like it's a constant struggle to keep up because of lack of training, this frustration may turn into a reason to leave. Giving training is a good investment as it allows your employee to be more productive quicker than if he had to learn something from scratch. An important side benefit to his productivity is his feeling of putting to use what he learned and satisfying his thirst for knowledge. An inexpensive and productive way to do this is to provide in-house training by one of your own "experts". This idea cuts costs and recognizes a star employee all at the same time.

With all of the latest technology bombarding the HP market, a common complaint by employees is not being trained on the "latest, greatest". Employees want to stay current in their knowledge of technology and resent working on out-of-date equipment or unsatisfactory software. They know it's career suicide to do otherwise. Employees will jump ship just for the opportunity to work in a "state of the art" shop. As managers, staying current yourself with what's new and desirable on the technology front will keep you ahead of the game, make your shop an attractive place to come to and one that people will not want to leave.

"I'm stuck with a boss that hates me". What do you do when you have a personality conflict? IBM Corporation gives "skip-level interviews". This is where the boss's boss meets with the employees to hear what's going on. One benefit of this is for the senior manager to "hear from the horse's mouth" what's going on in the department, rather than only what the department manager wants to report. A critical benefit for the employee is that he has "an ear" for his complaints and an outlet for his frustration. Second-level management reviews of performance appraisals work well too. This is where the boss's boss sees the appraisal before the employee does, and then again after the employee is given the appraisal, but before it goes to Personnel. This offers good objective safeguards.

What if you have a "bad apple" who's driving everyone away? We've all seen this happen, and it always seems to take numerous sacrifices of excellent employees before management wakes up and does something about the "bad apple". Bad employees add to the work load of the good employees, and the good ones end up being "punished" for someone else's failures. This is sad to see, and it's often symptomatic of the manager's inability to deal with problems. As a manager, you can prevent this from happening by being decisive and cutting the rope when necessary. Keeping someone on board who spoils the rest of your staff is only going to cause you more problems in the long run. No one is indispensable and even if this "bad apple" is the most knowledgeable person you have on staff, cut your losses and do it quick! Remember, it's not the people you fire that give you trouble, it's the people that you don't fire.

Is your shop an unhappy place to work? Make it fun! Work does not have to be serious, and laughter cures all ills. I've seen more contented employees in companies just because it was a fun place to be, and everyone had a good time working there. Be creative! Start your next meeting with a video of "baseball bloopers" to get everyone in a good mood. And who says that a department meeting has to be in a conference room? Take everyone out to the park and have them bring their lunch. If you're in a coat and tie environment, have "casual day" every other Friday. On your way to work, stop by the 7-11 and pick up the big 50 cent bubble gum. That will turn everyone into kids when each person compares how big their bubble is. Dress up for Halloween. Have a team golf or volleyball game. An owner of a company once said to me, "I feel that I'm in the entertainment business-making my people happy." As manager of your department, you can make this happen too.

$$$$...Salary is a common reason given for leaving although I believe that it is generally secondary to the other reasons. Sometimes you just have to dig deeper to uncover the real reasons. Too often they only come to light at the exit interview when it's too late to remedy the situation. If someone is unhappy with his job, he may say to himself.."I'm not paid enough to take this abuse". But I've seen too many times where an employee will stay with a company for a long time, underpaid, simply because he is truly happy with his working conditions, treatment and job functions. However, let's address the salary issue. Is pay based on performance more than on seniority? It better be or you'll end up keeping your oldest employees instead of your best ones. Have you been staying current with the going salaries in the marketplace or have you fallen behind? Has the employee's job grown to where it needs to be reevaluated and a new worth established? Assess each job objectively and decide what it is worth. Then pay accordingly! An added possibility is the incentive bonus. I have a client who uses the incentive bonuses very successfully and reports a dramatic increase in productivity. He and the employee set concrete and clear goals at the beginning of the year, with several goals (8-10) to reach and then monitors the progress monthly. He says the employee has a sense of pride, ownership and reward when the goal is achieved, and the money turns out to be only a side benefit.

Let me leave you with a story. There was this data processing company that was seeking a technological breakthrough which would put their company out in front in their industry. Because this was a make or break situation for this company, the entire staff was hard at work to achieve this goal. One day the President of the company was in the lunchroom having lunch, when one of his technological genius's suddenly found the answer to the problem. He had broken through the barrier and discovered what they were all looking for! The President was obviously thrilled and excited beyond belief, and wanted to instantly reward this employee for this achievement. In his excitement, the only thing he could think of saying was, " Here. Take my banana!" Well, the employee was so pleased with being recognized by the President, that he put the banana on his shelf for all to see. It turned brown, started to smell, but he wouldn't throw it away. When the President saw what the employee had done with the banana, the symbolism hit him and he immediately went out and bought little banana pins. He went back to that employee and pinned the little banana pin on him. From that point forward, the 1 cent "Top Banana" pin became the badge of highest honor in that company.

The moral of the story is corny but real...don't be cruel to a "heart that's true" if you want to keep your "top bananas"!

INDEX TECHNOLOGY VERSUS RELATIONAL; A USER'S DILEMMA

Presented by:
Daniel R. Filkins
DYNAMIC INFORMATION SYSTEMS CORP.
910 15th Street, Suite 640
Denver, CO 80202
(303) 893-0335

The purpose of this paper is to discuss in understandable terms the pros and cons of Index Technology versus Relational Technology in the Hewlett Packard (HP3000) computer environment.

Today's user is faced with the decision whether to embrace new technology, or continue to use older software at the risk of becoming antiquated and missing the possibility of dramatic performance improvements.

Index and Relational technologies are relatively new in the general sense. Both offer performance improvements, but because of their inherent architectures, vastly different performance gains are obtained. The user needs to pay attention to these differences.

Some of the questions answered by this paper are: 1) Is Index or Relational technology cost effective? 2) What are the benefits. 3) What are the pitfalls involved?

After reading this paper the user will be able to distinguish between Index and Relational Technology, and will be able to help their company move forward into the future.

THE EVOLUTION OF DBMS TECHNOLOGY

Modern Data Management Systems (DMS) technology started in the 1940s with the advent of the computing machine. Data management was not sophisticated at that time, rather it was in reality an expensive use of an electronic filing cabinet in combination with punched cards and/or magnetic tapes.

Data was at best scattered, repetitive, inconsistent, private, uncontrolled, inaccessible, time consuming, and in some cases unorganized. At that time, it was the best available, but from today's perspective defining it as a headache would be a gross understatement!

HIERARCHICAL DBMS

After years of struggle, free and forward thinking data processing wizards decided there really was a better way. First generation technology utilizing flat files (cards and tapes) was not meeting current needs. What evolved was a sophisticated Data Base Management System (DBMS) known as the Hierarchical DBMS.

A real breakthrough had occurred. Data could be controlled, shared, secured, retrieved, updated, and guarded for consistency to a certain degree. Highly paid technologists used sophisticated algorithms to control physical storage of the data, as the orientation was towards data location. Data was perceived in a very specific manner, and no other data perception was allowed.

Hierarchical systems stored data as if it was a "family tree". It was relatively easy to process the data, provided you happened to be a genealogist and had the "knowledge". The flaw in the hierarchical theory is that in real terms, data is not normally tree-organized as most everyone knows.

Along came the circumventors, another group of data processing specialists working on a higher plain. Their main job in life was to use the hierarchial DBMS for a purpose in which it was not intended. These highly paid specialists were able to perform near miracles as they went about developing systems knowing that there had to be something better than a hierarchical DBMS. The limitations were many. A hierarchical DBMS is very complicated to use, system design is complex, and no end user facilities are available. Refer to Diagram 1 for a hierarchial data structure example.

NETWORK DBMS

Alas, another breakthrough happened! This time the new technology evolved much quicker than before. The Network DBMS came upon the scene. There was rejoicing in the streets! The one-parent-to-many-children concept so heralded in the hierarchical world was enhanced with the one-child-to-many-parents concept. Restrictions seemed to disappear overnight, although data storage continued to be the main focus as the data processors attempted to make maximum use of scarce resources. This resource scarcity was also a limiting factor with hierarchical systems. The hierarchical and network systems used empirical data structures requiring a large number of empirical operators for what turned out to be incomplete operational ability. Refer to Diagram 2 for a Network data structure example.

With each iteration of newer and better DBMS software systems came more functionality, data base management utilities, faster access, multiple access methods, less restrictions, and so on. Network systems would soon reach their "heyday" and become the industry standard.

New technological developments were evolving at a quicker pace. Faster computers and cheaper Direct Access Storage Devices (DASD) were the norm rather than the exception. All of a sudden, the general populace discovered that there really was more data to store than was originally thought. A great cry came up from below! The users required flexible and friendly ad hoc access to their data, application systems were needed now rather than tomorrow, and by the time tomorrow came there was another need to be met. The Network DBMS was not meeting the needs of the user.

The data processing specialists worked hard and long hours. As hard as they tried, the data processing department could not get the Network DBMS to be very user friendly and flexible. There were too many design constraints holding the users back from their true calling. Despite heroic efforts the jealous gate keeping data processing guardians could not put in enough overtime hours to meet users demands. Outstanding user requests continued to build with backlogs sometimes exceeding one, two or more years of work. A great plague was set upon the data processing department.

Solving the dilemma of providing data to users in a friendly and flexible manner is a software builders dream. New high powered languages and other types of technology came to the market. As fast as the new technology became available, users kept at least a few steps ahead by demanding that independent of how the data was stored, there had to be a better way of presenting the information that was so desperately needed.

RELATIONAL DBMS

Circa 1970, the father of the modern relational model, E. F. Codd, started work on relational technology. The relational model is a well established mathematical concept applied to data. Origins of the relational model can be traced back to the ancient Greeks.

Simplistically, the relational model as we see it today is data represented by a collection of tables. A table has "columns" (vertical) and "rows" (horizontal). At the point where a column intersects with a row the resultant is a data "field". Within the field resides a data value. Data is stored in "tuples" (otherwise known as pages or blocks). Refer to Diagram 3 for Relational data structure example.

The Relational Data Base Management System (RDBMS) operates using "set theory" whereby a resultant select operation (find) returns all the data located that meets the criteria. The one-record-at-a-time concept no longer applies. Relational systems use SQL commands such as selects, projects, unions, intersections, differences, products, divisions, and joins. Refer to Diagram 4 for examples. Incorporated within most relational systems is a B-tree indexing facility which allows for indexed access to user data. Refer to Diagram 5 for B-tree example.

With this newer technology came a different way of doing things. There were new terms to grapple with such as domains, attributes, primary keys and foreign keys, referential integrity, relational algebra, outer theta joins, and many more. Applications now required that data be normalized typically to the third normal form.

Briefly, data normalization is as follows: First Normal Form is where there are atomic domains. Every item represents a single fact. Second Normal Form is an enhancement to the first normal form plus the elimination of partial dependencies. Third Normal Form is a further qualification of the second normal form plus the elimination of transitive dependencies. Every row represents a single fact.

Relational systems, although thought to be simple to understand and apply have become a highly complex and complicated model to implement. Because implementing relational technology is very different from what a typical data processing shop now understands, there is a great need for additional education, additional time to re-design existing systems, additional time to tune systems, and other activities associated with successful implementation. It is worth noting that extra time should be spent up front on the design stage. The dividends will be paid only if the design is handled properly. Program development is now only a small part of the development process.

In addition, RDBMS applications can be slow performers in many cases. They are definitely easy to use, flexible, and offer portability between hardware platforms, but performance in the past has been a secondary issue

For an RDBMS to be truly effective a great variety of tools are required. Those tools are the form generators, the menu generators, precompilers, report writers, ad hoc query facilities, SQL code, graphing, spreadsheets, and other functions.

With so much to learn and so little time to get systems into place to meet current needs, it is no wonder that within segments of the user community a user revolt is beginning. Users want high performing systems today, not tomorrow. It has been twenty years since establishing the foundation of the modern relational model (E. F. Codd, circa 1970). One begins to wonder what really has happened over this span of 20 years. Is it over-kill?

Technology has certainly improved time and again. Hardware performance has greatly improved, there is a plethora of software available, and users still need immediate access to their data. Why is it then that a large majority of the data processing community has not readily embraced relational architecture, the supposed panacea for all problems?

Is it because RDBMS systems can only support small environments? Is it because RDBMS systems are for non-production applications? Is there a lack of understanding? Is RDBMS a fad? Why is there a group of hardware manufacturers jumping into the RDBMS fray and introducing back-end processors to speed up access to the RDBMS data? What is the answer?

INDEX TECHNOLOGY, THE COMFORTABLE SOLUTION

In the late 1970s and early 1980s a small group of technologists realized that relational was good in theory, but applying it in a practical sense may not be profitable. With this in mind, it was ascertained that there may a better solution, that of INDEX TECHNOLOGY. Refer to Diagram 6 for an Index Data Structure example.

Users have always wanted to have their cake and to eat it too! Why not offer them what they want, performance, and also give them ease of use and flexibility. To provide this of the user, Index Technology became a necessity.

There is one indisputable constant in the computer world: Fast is never fast enough. The main focus of Index Technology is high speed access to the data a user needs, in addition to providing relational like flexibility and ease of use. High performance retrievals are based upon sophisticated indexing algorithms. The indexing aspect is one of the most important facilities relating to database design and implementation. Sophisticated indexing is a very effective way to reduce disk I/O required to retrieve a subset of data. Although CPU speeds continue to increase as semiconductor technology evolves, mechanical disk drives are still limited to about 30 I/Os per second. It is understandable then that it is critical to reduce I/O because a majority of application systems running on the CPU continually wait for data from the disk drives.

In conjunction with reduced disk I/O, good disk locality is critical to the success of high performance systems. Good locality means that as a process is requesting records, there is a likelihood that records in close proximity could be retrieved with the same I/O. These records can then be accessed in the memory buffer without requiring additional disk reads.

With advanced indexing techniques record selections by any criteria can be accomplished using reduced disk reads. This permits complex selections from large databases to be completed in seconds. Another important item in a computer environment is that of sorts. Sorts are so important that all hardware vendors and many third parties provide high-speed sorting capabilities. With Index technology a sorted key is available thereby allowing the user to retrieve data in a sorted manner without having to exit to an untimely sort routine.

The user needs an unlimited number of keys to access the data, and the keys must be available in many different flavors such as partial keys, composite keys (parts of keys, or non-contiguous combined keys), range keys, and soundex keys. What the user does not realize is that there is a need for reduced I/O through the elimination or reduction of serial reads to the data files. Like a relational system Index technology utilizes a B-tree index structure to provide keyed sequential access to the data. B-tree indexes provide several additional capabilities including the ability to select a range of data and the ability to use partial keywords as search criteria all the while using relational operators.

Given the power of B-tree indexes, one would assume that the B-tree is the state of the art in database access technology. Not so! Index technology has gone beyond B-tree indexing by providing two additional features: keyword retrieval and high speed, multi-field, multi-table (set) selection. Keyword retrieval is the ability to retrieve records or documents by any combination of words and values they contain. When a field contains several words, such as company name or product description, any of the words can be used regardless of their position in the field. Multi-field, multi-table (set) selection extends that capability across multiple fields and tables (sets).

Through the use of Index technology and the advanced index structures, a user can qualify records at the rate of approximately 10,000 records per second per keyword (selection value) regardless of the size of the table (set). Indexed retrieval can be up to 10,000 times faster than a serial read.

Index Technology as it is today in the Hewlett Packard (HP3000) environment is a marriage of the best parts of the Network world and the Relational world. Millions of dollars have been invested in Network technology offering high performance production systems. These systems are mature and profitable. Millions of dollars have also been spent on advanced index technology.

Although indexing is not the solution to every database performance problem, Index technology is here to stay. There is a cost/benefit tradeoff. Although disk I/Os are required to maintain an index, disk I/Os are saved when the data is retrieved. Index technology translates into performance, flexibility, security, stability, reliability, and profitability.

The value of your data is only as good as your access to it. To provide an additional understanding of Index technology the following compares Index and Relational technology.

INDEX VERSUS RELATIONAL TECHNOLOGY

Index and Relational technology both provide the following functionality. Each type of unique architecture offers amazing similarities, yet the differences are recognizable.

- Simple and flexible data base design
- Simple retrieval and data manipulation
- Security
- Easy user access to data with multiple "views"
- Easy maintenance of data integrity
- Data dictionary interface
- Boolean and relational logic at the table (set) level
- Unlimited keys
- Retrievals across tables (sets)
- Reduced developmental efforts
- Ad hoc inquiries
- Full set of DBMS utilities
- Interface to 4GL products

Index technology by itself offers many features and functions. The following is only a partial list. To detail a complete list of the unique facilities would take too much time to list here.

- Indexing structure independent from database design
- Very high performance indexed selections
- Minimized I/Os
- New keys without structural changes
- Partial key retrieval
- Sorted sequential access (B-Tree)
- Keyword retrieval on textual data
- Multiple keys at master and detail levels
- Range retrieval
- Limited structure knowledge required
- Interfaces to all 3GL programming languages

As is the case with Index technology, Relational technology has its uniqueness also. This is only a partial list noted in general terms.

- Data Independence
- DBMS structure is easy to maintain
- Limited structure knowledge required
- Full table scans
- User builds "relations" as needed
- Data clustering
- SQL interface
- Language Precompiler
- Interfaces to most 3GL programming languages
- Platform independence (portability)

Indexed vs. Relational Data Base Comparisons

Index/Network	Relational
Data are stored in tables (sets) that may contain repeating groups and compounded items.	Data are arranged in 2 dimensional arrays called tables.
Associations between entities (sets) are expressed through explicit structures and links (chains...).	Relations represent associations between entitites which are usually dynamic.
A hierarchical structure may exist (parent to child) explicitly between entities.	Any parent to child relationships within view are created dynamically.
All keys are predetermined.	Keys are defined dynamically for each view.
May have data redundancy to prevent insertion and deletion anomalies.	Insertion and deletion anomalies are virtually eliminated through a process called normalization and data redundancy may be kept at a minimum.
Designed for retrieval and update for implementation into production systems.	Designed for retrieval with updating as a secondary focus.
No portability between hardware platforms.	Built for portability between hardware platforms.

PROS AND CONS OF EACH DBMS

	Pro	Con
Hierarchical:	Performance Data integrity Security	Difficult to use Ad hoc limitations Difficult to design
Network:	Performance Data integrity Security	Difficult to Use Ad hoc limitations Difficult to design
Relational:	Simple design Easy to use Flexible Ad hoc inquiry Multi table (set) Multi field Platform portable Security	Performance problems Locking Design can be critical Overhead
Index:	Performance Simple design Easy to use Flexible Ad hoc inquiry Multi table (set) Multi field Security	Pre-defined keys

COST EFFECTIVENESS

Index and Relational technology can be very cost effective. This statement is made based on the fact that in today's competitive environment, a company can ill afford to remain in the dark ages with respect to computer technology. Your competition, whether you realize it or not, is automating at a rapid pace.

Do not let yourself fall into the trap that cyberphobics set. They tend to focus on short-term cost benefits and boldly declare that a system's costs is not equal to its perceived benefit. A company should also not spend more time cost-justifying software than using it to solve business problems.

Cost effectiveness can be measured two ways, tangible and intangible. Depending on the situation tangible effectivess could be reduced CPU utilization through I/O reduction, faster application response which frees the user for other activities, quicker implementation of application systems, more timely management information, and so on. Intangible benefits could be better employee morale due to the use of advanced technology, better quality employees, better quality work, higher levels of user satisfaction, and so forth.

Because data is a company resource and a company asset better access to corporate information makes for a more competitive organization. Money spent on information technology serves to protect the investment a company has in itself.

Movers and shakers spend as much money as possible on information technology. Even though this may appear to be paradoxical in tighter economic times, now is not the time to be conservative about investing in software. The 1970s are gone, and the line of thinking about grand designs has disappeared. In the 1990s companies must be able to use a new tool faster and better than the competition. Cutting back on information technology is comparative to cutting your own throat.

BENEFITS

The benefits of Index or Relational technology accrue to two areas. Firstly, the data processing department will recognize, depending upon the situation, increased throughput of the CPU. This could mean that jobs run faster, terminals may respond faster, and so forth. When building new systems, data base design is simplified. Systems can be put into place faster, more systems can be built in less time. Database maintenance (changes) are easier to make which means that as the environment changes, data processing can meet the needs on a timely basis.

Secondly, the user community receives recognizable improvements. Retrievals are faster, more timely information is available, "What-if" scenarios are feasible, access to data is not limited, and so on. Users can better do their jobs with fewer restrictions. The end user becomes more competitive against the opposition.

PITFALLS
Each technology has its pitfalls. For the purposes of discussion the pitfalls noted below apply only to relational and index technology.

Relational technology pitfalls:

- Over normalization of data structures can degrade performance.
- Application design can be extremely critical.
- A user can lock sets of rows or whole tables or groups of tables by using the select statement. Care should be exercised in using a select.
- In most cases there is often two ways to access data with SQL, a fast way and a slow way.
- Over time, an application system may require the addition of an index in order to offer a level of performance acceptable to the user. Full table scans may not provide acceptable performance.
- When using clustering be aware that too little data stored for each key (few rows per cluster key) may waste space, conversely, too much data per key may cause excessive chaining thereby reducing performance.
- Because a relational system uses an access optimizer there is a chance that the optimizer may select what appears to be the best method to access the data, but in fact the method selected could turn out to be the worst access method.
- Larger sized CPUs are normally required.
- Additional software tools are usually required.
- Additional training can be required since there are multiple products to learn.
- Reaching the stage of efficient usage of a relational system may require additional staff resources.

Index technology has these pitfalls:

- The user must know how he would like to access the data. Pre-defined keys are necessary.
- Like relational, when there are a lot of keys, the disk space required can be heavy.
- Also like relational, when there are a lot of keys, adding records can take an extra second or two.
- There is no portability between hardware platforms.
- To truly utilize index technology to a full extent it is best to redesign applications, not convert them.
- There are no triggering mechanisms available.
- An interface to SQL is not yet available.
- Binary and ASCII fields cannot be combined to create composite keys.

SUMMARY

When the final analysis is complete, selecting Index technology versus Relational technology can be pared down to several key factors.

Scenario #1: If there are very urgent performance needs to be met, then the logical choice is INDEX technology used in conjunction with the IMAGE DBMS.

Why?
- Index technology can be put in place quickly with a limited investment.
- Applications do not need to be converted immediately, the call conversion facility can be used.
- Existing hardware can be utilized.

Scenario #2 If there are performance needs along with user accessibility needs (not necessarily urgent needs) but there is a budgetary and resource restriction, then the logical choice is INDEX technology.

Why?
- Index technology can be put in place with a limited investment.
- Index technology provides easy user accessibility without causing heavy CPU consumption.
- Large amounts of staff resources are not required.
- Existing hardware can be utilized.

Scenario #3 If there is a need for data portability, end user access and flexibility, then the logical choice is Relational technology.

Why?
- Relational technology provides portability between hardware platforms.
- Relational technology is easy to use.
- Relational technology is very flexible.

Comment: What is not considered here are those organizations that want to implement Index or Relational technology for the sake of having the latest technology or to keep up with the Joneses.

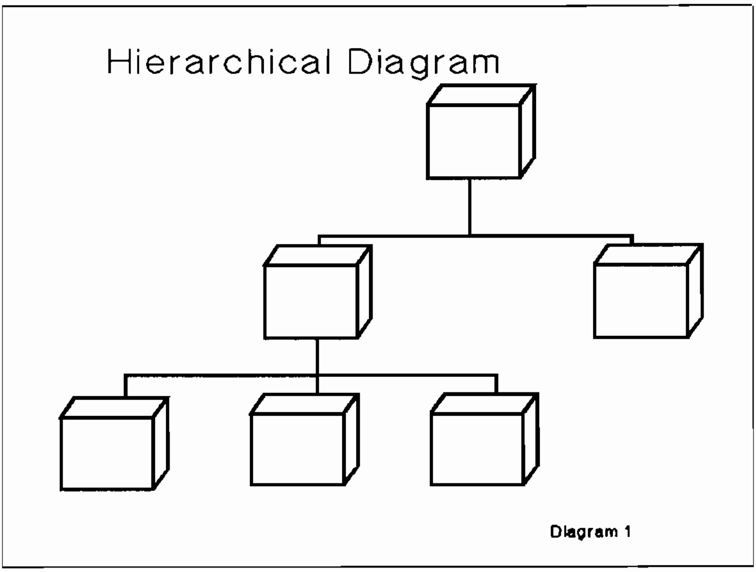
Caveat: Do not try to implemement 4GL/5GL technologies quickly if your organization is still trying to master second generation techniques.

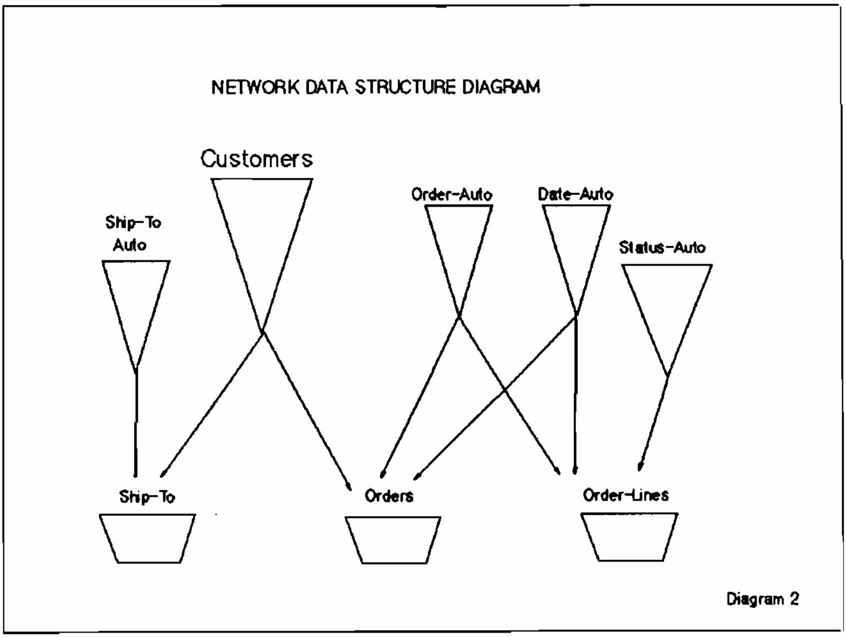
Definitions

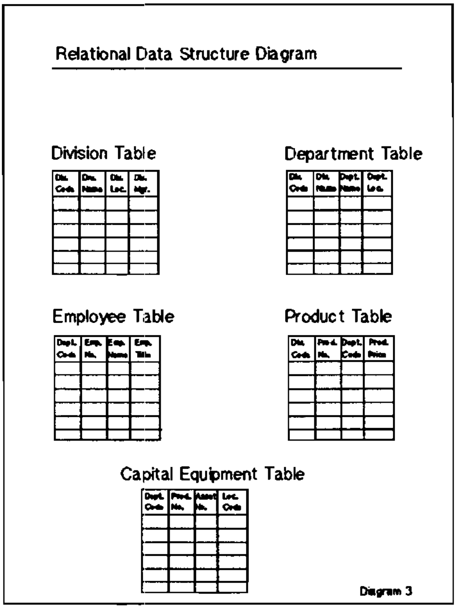
Attribute	A field containing information about an entity.
B-Tree	An index typically used by an RDBMS system.
Data Cluster	The physical storing of data near other related data.
Data Redundancy	The non-essential repetition of data elements throughout the relation schemes. Redundancy not only consumes valuable storage resources, it may also contribute to other types of anomalies.
Deletion Anomalies	The inverse of an insertion anomaly involving the potential loss of other essential information if certain rows are deleted.
Domain	A collection of data items of the same type in a relation.
Insertion Anomalies	The inserting of incomplete information.
Relation	A table; two dimensional array of data elements.
SQL	Structured Query Language
Table	A data file.
Tuple	A group of related fields.
Update Anomalies	A direct consequence of redundancy. Involves the failure of propagating an update of one row to other rows that require the same information. Data inconsistencies are end result.
View	Part of all of the database definition.

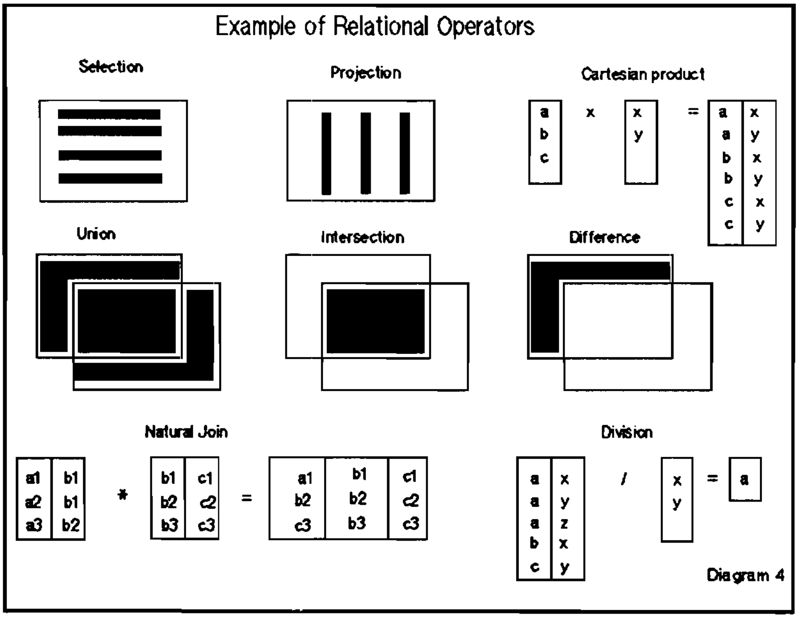
Relational Algebraic Operators

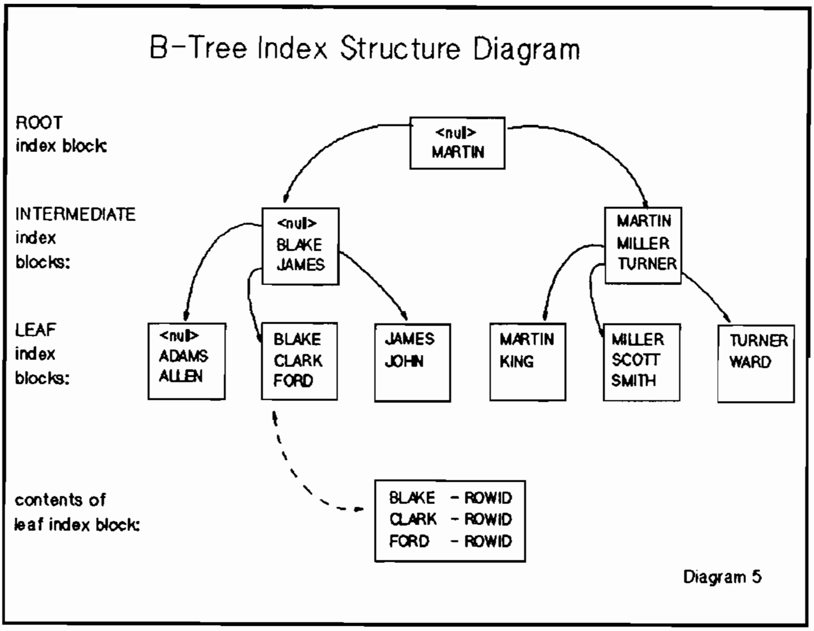
Cartesian	Allows users to create a new relation by concatenating all combinations of tuples from two relations.
Difference	Allows user to create a new relation from the tuples that are in one relation but not in another when both relations have the same definition.
Division	Allows users to create a new relation from the tuples of the remaining attributes of one relation where all the attributes in the tuples of another relation have a corresponding match with the same attribute tuples in the original relation.
Intersection	Allow users to create a new relation from the identical tuples of two equally defined relations.
Join	Allows users to create a new relation from the combined attributes of other relations when certain attributes from all the relations agree on a selection criteria.
Permutation	Allows a user to change the ordering of the attributes in the relation without affecting the contents.
Projection	Allows the user to define a new relation which contains a subset of the attributes from an existing relation(s).
Restriction	Allows the user to create a new relation which contains the tuples of one relation providing they satisfy the selection criteria involving tuples from another relation over an equally defined attribute.
Selection	Allows a user to define a new relation by specifying a predicate which selects entities from an existing relation(s).
Union	Allows users to create a new relation by the addition of unique tuples from one relation to another relation when both relations have the same definition.

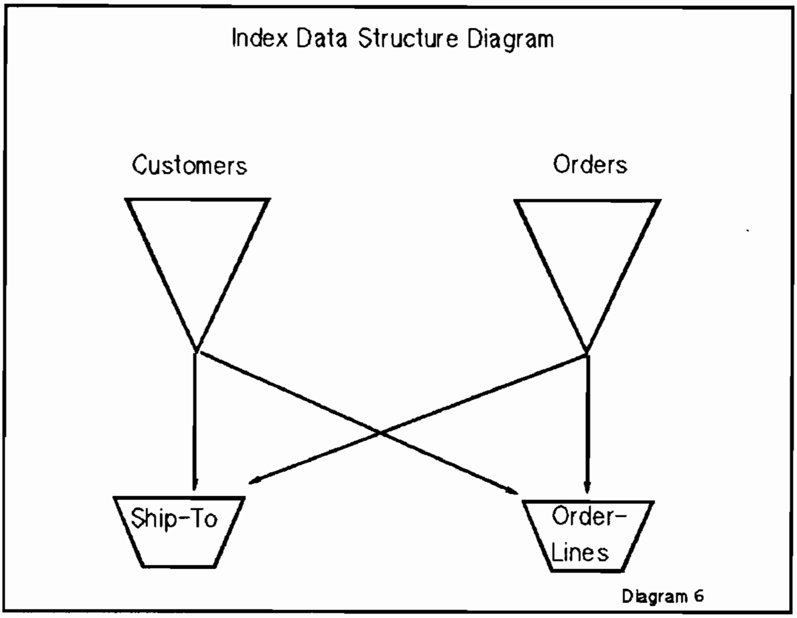

Hierarchical Diagram
Diagram 1


NETWORK DATA STRUCTURE DIAGRAM
Customers
Ship-To
Auto
Order-Auto
Date-Auto
Status-Auto
Ship-To
Orders
Order-Lines
Diagram 2


Relational Data Structure Diagram
Division Table
Department Table
Employee Table
Product Table
Capital Equipment Table
Diagram 3


Example of Relational Operators
Selection
Projection
Cartesian product
a
b
c
x
x
y
=
a x
a y
b x
b y
c x
c y
Union
Intersection
Difference
Natural Join
a1 b1
a2 b1
a3 b2
*
b1 c1
b2 c2
b3 c3
=
a1 b1 c1
b2 b2 c2
c3 b3 c3
Division
a x
a y
a z
b x
c y
/
x
y
=
a
Diagram 4


B-Tree Index Structure Diagram
ROOT
index block:
INTERMEDIATE
index
blocks:
LEAF
index
blocks:
contents of
leaf index block:
<null>
MARTIN
<null>
BLAKE
JAMES
MARTIN
MILLER
TURNER
<null>
ADAMS
ALLEN
BLAKE
CLARK
FORD
JAMES
JOHN
MARTIN
KING
MILLER
SCOTT
SMITH
TURNER
WARD
BLAKE - ROWID
CLARK - ROWID
FORD - ROWID
Diagram 5


Index Data Structure Diagram
Customers
Orders
Ship-To
Order-
Lines
Diagram 6

DBMS Comparative Definitions

Network	Indexed	Relational
Account	Account	User (set of tables)
Database	Database/logically related tables	Logically-related tables
Dataset (file)	Entry/Row	Table
Entry (record)	Item/Column	Row
Item (field)	Path/Relation	Column
Path	Key/Index	Relation
Key		Index

Diagram 7

DBMS Systems

Performance vs. Flexibility

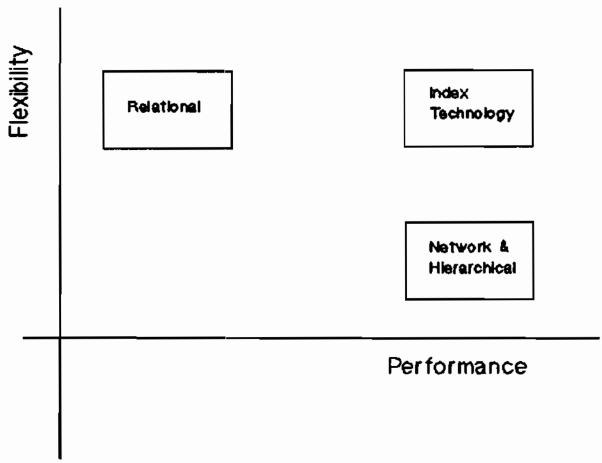
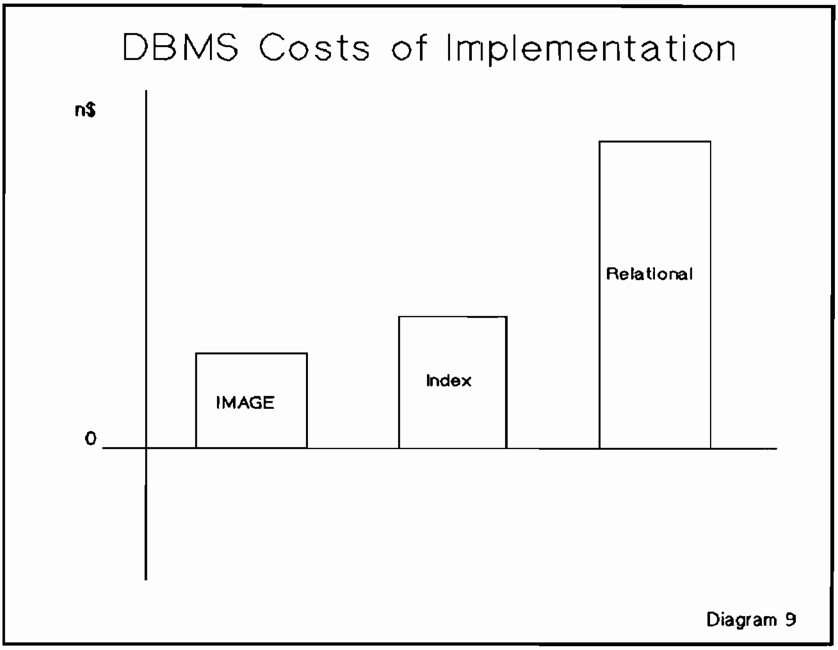


Diagram 8


DBMS Costs of Implementation
n$
0
IMAGE
Index
Relational
Diagram 9

TITLE: Information Management - The Corporate Weapon of the 90's

AUTHOR: Robert Andrews
Dynamic Information Systems Corp.
910 15th St., Suite 640
Denver, CO 80202
303-893-0335

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 5200

Interactive Computer Graphics in Business

Jerry M.C. Kopan
Hewlett-Packard (Canada) Limited
Toronto, Ontario, Canada

Abstract

Humans think in a very colourful and abstract way, using words, numbers, pictures and other complex techniques. A computer on the other hand, "thinks" in a black and white fashion, using bits, bytes, voltages and other absolute or discreet techniques. To a human a graph is an efficient and easily understandable method of describing a relationship between several variables. A computer must store thousands of bytes of data before it can analyze and display the relationship for the same variables. You could say:

> "To a human a picture is worth a thousand words,
> to a computer a picture is worth a thousand '**words**'"

Computer graphics is a relatively new technique that is breaking through this human interface barrier. Computer manufacturers and software vendors have realized the worth of interactive computer graphics to the businessman. The adoption of X Windows and OSF/Motif is an indication of this trend. Open Systems will make it easier to move into the graphics arena. Interactive computer graphics is revolutionizing the role of the computer and the way humans interact with them.

Introduction

Interaction with computers has always been a major stumbling block which has drastically reduced their efficient use. The introduction of bit-mapped graphics monitor has made it possible to display an application in a pictorial way to a user.

Being more profitable is the aim of business. In the mid 1980's Apple made graphics very popular with the Macintosh. Apple developed software packages that were easy to learn. The use of consistent graphical interfaces made it possible to decrease the time required to learn how to use an application. A consistent user interface implies a common look and feel for an application. Hence the learning curve is reduced, people become more productive and finally this results in a higher profit margin for business.

A major hurdle that graphics had to jump was the CPU bottleneck. Graphics is CPU intensive. As CPU performance increases and costs decrease graphics becomes more and more attractive. Typically graphics has been used in arenas which produce a high return on investment.

The automobile, military and aircraft industries were the flag ships in pioneering the field of computer graphics. Back in the "old" days these applications ran on "mighty" mainframes and the financial investment was immense, but there was a return, such as higher profits, greater safety and faster design cycles. Today we can run powerful CAD applications on desk-top computers. The computer industry has changed!

People tend to think that the graphics adds a layer of complexity to an application, this need not be the case. Application designers should use the power of computer graphics to simplify complex formulas and data representations.

The steady trend towards standards and open systems has cut a path for computer graphics into the business arena. Open architectures, scalable systems, and multi-vendor compatibility are the new directions for the future.

Many PC based software developers use MS-Windows to take advantage of the graphics capabilities of systems. Hence MS-Windows has become a defacto standard for PC based graphics applications. Apple has standardized all applications on their windowing system.

The Open Software Foundation (OSF) has chosen the best features from several vendor environments to create Motif as the look and feel for applications. UNIX International has chosen Open Look as the look and feel for applications.

There is a definite direction towards standardization of graphical interfaces so that applications can be written that are easier to understand, learn and use. As graphics standards are further enhanced, it will become as common place as terminals are today.

We can benefit from the pioneering work that the billion-dollar companies have uncovered for us. To be more productive and profitable business needs to incorporate interactive computer graphics.

Some Typical Business Application Using Computer Graphics

Different types of business are taking advantage of the the power of computer graphics. Below is a list of such business areas.

Cartography

Computer graphics is used for the production of highly accurate representations on paper or film of geographical maps, relief maps, exploration maps for drilling and mining, oceanography, weather maps, contour maps, oil exploration maps, and population density maps.

Computer-Aided Drafting and Design

In computer-aided design (CAD), interactive graphics is used to design components and systems of mechanical, electrical, electromechanical, and electronic devices. These systems include structures (such as buildings, chemical and power plants, automobile bodies, airplane and ship hulls, ...), optical systems, and telephone and computer networks. The emphasis is on interacting with a computer-based model of the component or system being designed in order to test, for example, its mechanical, electrical or thermal properties. Often the model is interpreted by a simulation program which feeds back the behaviour of the model to the user for further interactive design and test cycles.

Simulation and Animation

Computer-produced animated movies are becoming increasingly popular. We can model extremely complex formulae such as hydraulic flow, relativity, nuclear and chemical reactions, physiological systems and organs, deformation of structures under stress, and walking. By simulating these models we can observe changes without the expense, time or safety of human life.

A relatively new and also high-technology area is interactive cartooning; such cartoons have a very high visual quality and it is becoming more cost-effective to use sophisticated computer graphics technology to eliminate routine steps.

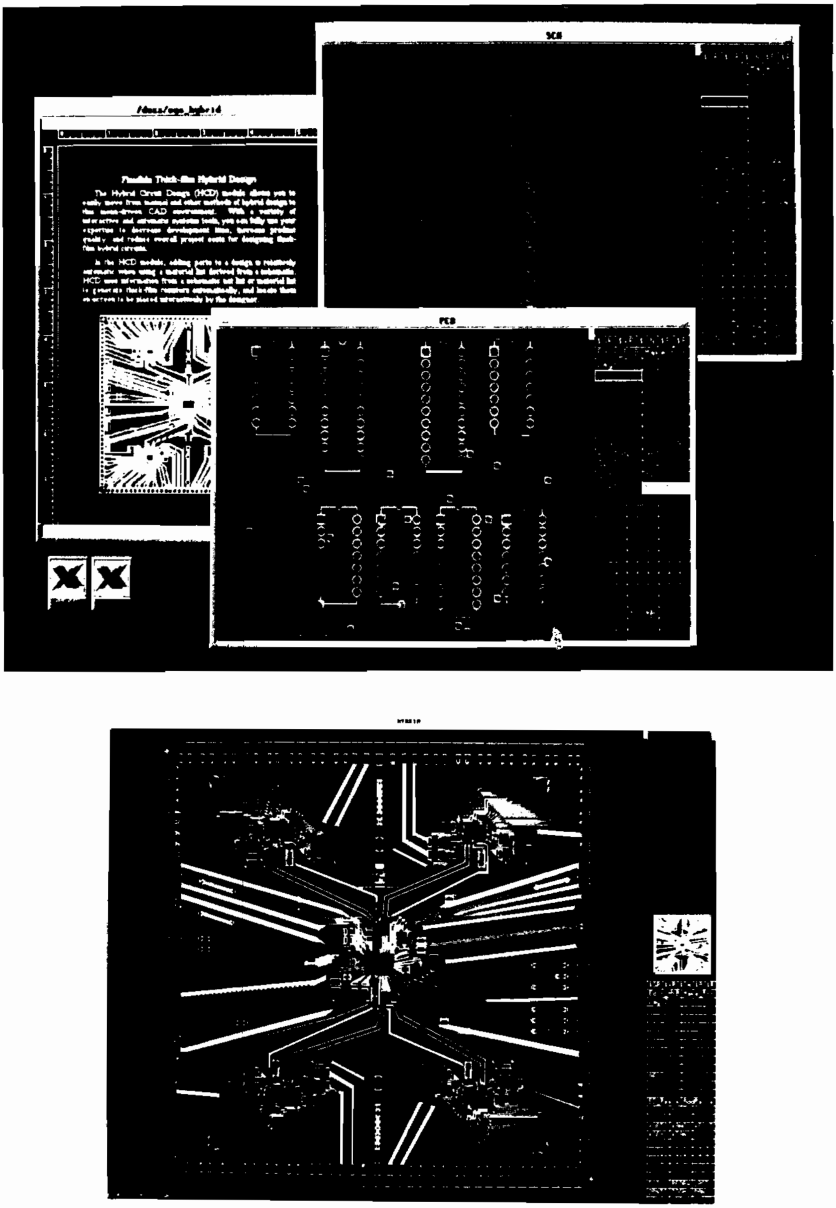


Figure 1. Engineering Graphics System (HP EGS) - Two dimensional graphics software for electronic, mechanical, and general engineering applications.

There is a very short life span of most companies in the animation industry. As in any business the cost of production and operation is passed to the buyer. In the early days of computer animation the work was being done on very expensive systems, suchs as DEC VAX 780's with a 1 MIP performance rating. These systems were selling for between $100,000-$200,000.

Amortizing the cost of these systems over several years it became very expensive to operate as a business. The cost of hardware was decreasing at an incredible rate. Today the cost of 1 MIP is between $500-$1000. Businessmen were still paying off the cost of their original systems, and passing these costs to their clients. As you can imagine new companies emerged to sell their services at a much lower rate.

An interesting phenomena has taken place in computer animation and visualization. In the early days it took 2 hours to render an image, today it still takes 2 hours. The reason for this is that as rendering systems have become more sophisticated, the designer is becoming more demanding and hence producing more complex images. To metaphrase Bob Able of, now defunct, Omnibus:

> "A picture may be worth a thousand words,
> but if you can make it move it's worth a million"

<u>Process Control</u>

Refineries, power plants, galvanizing mills, and computer networks display data values from sensors attached to critical components in the system; the operator needs to respond to exceptional conditions. The Telephony industry uses Dynamic Control Routing to switch to alternate paths in cases where there is too high a load on one path. This new path is displayed graphically to the operator as the new routing takes place. As the load on that one line is returned to normal, lines are re-routed to ensure that a balanced environment is maintained. In a sheet metal galvanizing mill feedback from sensors determines the thickness of galvanized zinc. As the thickness increases, air jets are used to reduced the thickness of the zinc as the sheet metal exits the molten zinc pot.

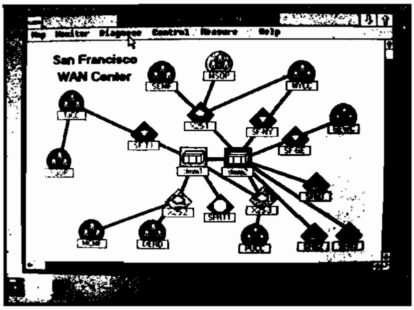


Figure 2. Open View - Graphical Network Management Tool.

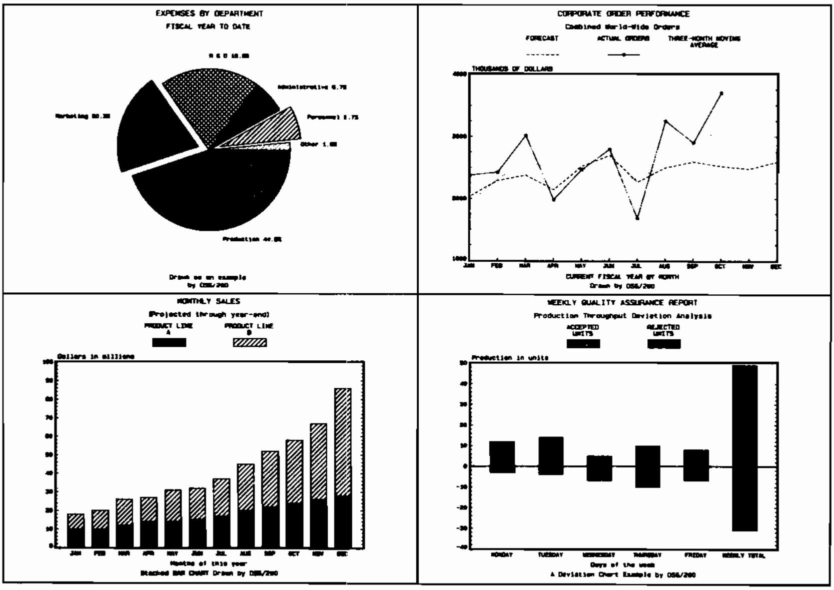


Figure 3. Example of intuitive and easy to understand reports that were generated using an interactive graphics package.

Office Automation and Electronic Publishing

There are several important requirements for applications in the office.

1. It should be easy to use. This is usually achieved by providing a menu to aid diagram design and selection. Using a menu a user would only have to point to a particular selection and option. This eliminates having to use a document to find a command or option, because all your commands and options are clearly marked on the screen.

2. Economic. It should be cost-competitive with other conventional applications.

3. Simple to update and change data. This is especially important in an interactive environment where multiple solutions to a problem can be tested. It is important in keeping statistics up to date.

4. Rapid and flexible. Again, where executives are in a meeting it is vital that the system should be quick to respond to requests for a variety of diagrams.

Electronic publishing is used to generate documents which have words, tables, graphs, and pictures. Displaying this type of document is very difficult on a dumb terminal. With the use of interactive graphics we can display a what-you-see-is-what-you-get (WYSIWYG) type of user interface. This reduces the time spent on the aesthetics in a document, which is typically 20 percent. Before the WYSIWYG approach scissors and glue where very much a part of this environment.

Art and Commerce

Computer art and advertising have the common goal of expressing message and attracting the attention of the public with aesthetically pleasing pictures. Very sophisticated mechanisms are available to the creator of the picture for modeling the objects and for the representation of light and shadows. Finally, the production of slides for commercial, scientific, or educational presentations have become a very cost-effective use of graphics, given the steeply rising labor costs associated with traditional means of creating and processing such material.

Interactive Computer Graphics in Entertainment

"Captain Power and the Soldiers of the Future" was a children's show airing on Saturday afternoons. This production married live action characters and scenes with 3D animated and rendered characters, such as Soaron and Blastar. The fact that there were live characters interacting with 3D animated characters made it very appealing and won a large audience.

But what made it unique was the interactive computer-generated visual signal that was embedded into each frame. This interactive signal was then used to interact with store-bought "Captain Power" guns. A player sat in front of the TV and aimed and fired at the characters on the screen. Depending on the type of signal the gun registers a hit for the player or against the player, if for example Soaron fires and hits the player.

This project raised many eye-brows. There were several people who praised it for its technical merit. The technology used in this project won a Gemini Award (Canada's Emmy) for Best Technical Achievement.

On the other hand there were several people who were against the show. Several parent groups were outraged with the show, stating that their kids were already exposed to too much violence and were spending too much time in front of the TV. There was also criticism about the producing a show for promoting the sale of "Captain Power" toys, much like several shows on Saturday morning TV are doing today. The Captain Power paraphenalia did not sell very well and hence resulted in the cancellation of the series.

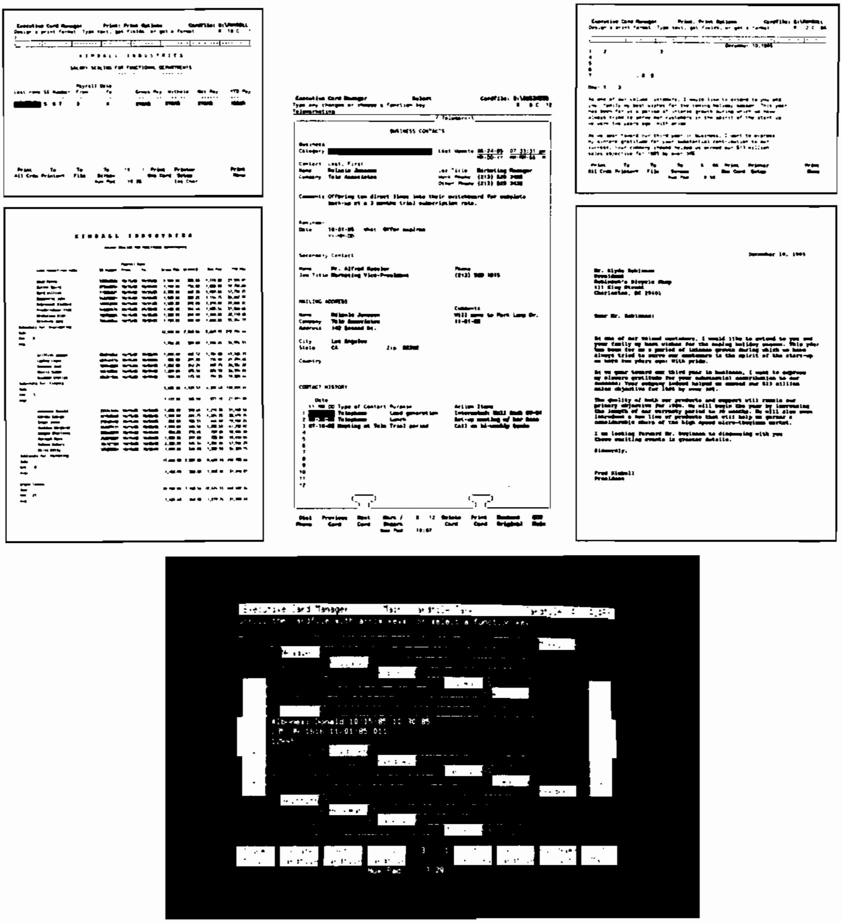


Figure 4. Executive Card Manager maintains the distinctive Rolodex cardfile interface.

Open Systems

Open systems will have a profound impact on the success of interactive computer graphics. The proprietary days for software development are over. Today, Graphics developers want to protect their investment of time that they have put into developing their applications. Such standards as Phigs, GKS and X11 are replacing proprietary solutions.

In an open systems environment, the computer systems and the software vendor are interchangeable and can be combined into an integrated operating environment. Some terms you'll often see associated with open systems are portability, interoperability or interconnectivity.

There are several standards committees, but two notable bodies are Open Software Foundation and UNIX International. These two organizations are responding to user's demands for the timely introduction of a common, standardized environment. These two bodies are promoting a graphical user environment for the look and feel of applications, such X Windows.

The principle concept of the graphical user interface is to allow people who use computers to establish a consistent behavior, without regimenting appearance. Thus this look and feel describes, how a push button or a menu "works", how the push button or menu "looks".

The reason for this distinction between behavior and appearance is simple. A person's ability to drive two different automobiles doesn't depend on the colour of each car, their shapes, or the number of seats. It depends on common behaviour: the steering wheel steers, the accelerator is the pedal on the right, and the brake is on the left. Thats not to say that two standards can't exist, such as automatic and standard drive cars, both having different features.

Similarly, a person's ability to easily learn and operate different programs is based, not on common appearance, but on common behavior between the programs.

A typical response to the question, "How do I copy field 1 in application A to field 2 in application B?", is "I don't know how application A works, lets wait until tomorrow for the secretary to do it for us". Incredible, eh!

Using a common graphical environment, people can move among systems, unaware of the differences that exist in the underlying technologies.

Standards in interactive graphics is significantly enhancing the usability of computer systems.

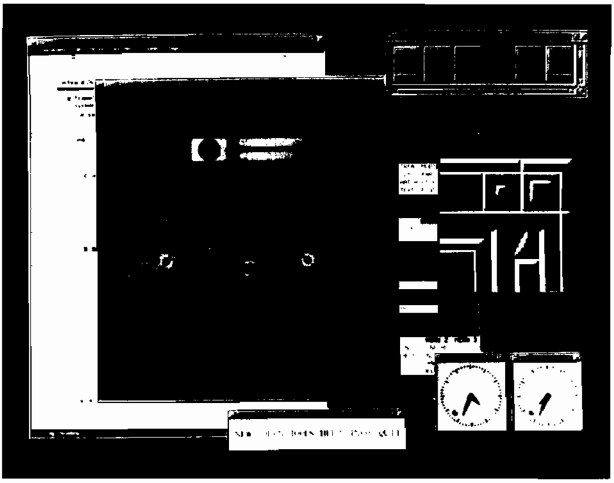


Figure 5. X Windows

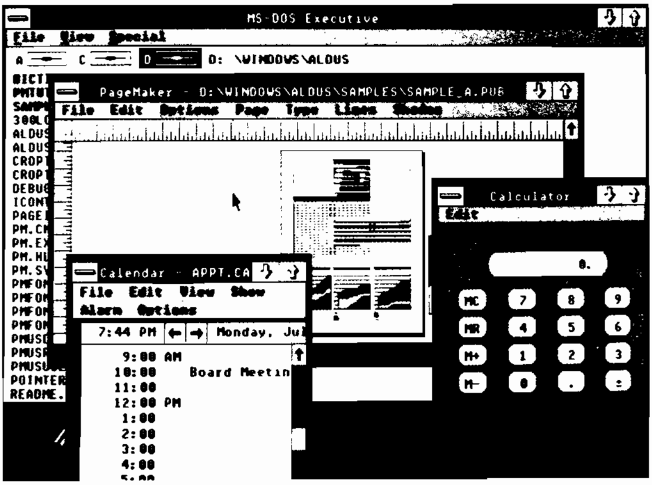


Figure 6. Microsoft Windows

The Future

Computer graphics has too often been considered a special form of communication requiring special interfaces and therefore only used when needed. Graphics in applications has always been cited as an exotic extension, it requires additional processing, storage and a high-quality display.

Fortunately, the decreasing cost of hardware is making interactive graphics more feasible. Interactive graphics is becoming the medium of communications between computers and humans. Today many courses use interactive plotting and modeling or simulation on low-cost graphics terminals attached to computers.

The use of broadcast television and/or the telephone system, plus a simple keyboard or mouse, to let TV viewers select items from customized "electronic newspapers", browse through on-line encyclopedias and yellow pages, and get stock market quotations, entertainment listing, etc. Much of the information is transmitted in graphical form. Once this medium becomes sufficiently reliable and cheap, using a computer-connected TV as an information resource for text and graphics will become as natural and commonplace as using the telephone.

The fields of office automation and word processing are introducing numbers of office workers to computers for document preparation, electronic mail and other office functions such as spread-sheets and database management systems. The demand for tables, charts, forms and figures is increasing. Hence systems are expected to display them. This "what you see is what you get" design philosophy, in which the screen mirrors the printed page eliminates the bothersome and unnatural formatting or typesetting codes whose effects are not seen on line.

Telemarketing will emerge into the 90's as it will take advantage of interactive graphics. We'll be able to view houses in our homes, and be able to select different districts or cities.

Datacommunication channels are getting wider as new technologies arise. This means a broader bandwidth for data to flow through. A broader bandwidth means we can have multiple channels, hence many users. This wide bandwidth will make graphics more common because the demand will increase. This is only the beginning of interactive computer graphics.

Bibliography

J.D.Foley and A. van Dam, Fundamentals of Interactive Computer Graphics, Addison-Wesley Publishing Company, 1982.

Bruce R. Dewey, Computer Graphics for Engineers, Harper & Row Publishers, 1988.

Dennis Harris, Computer Graphics in Applications, Chapman and Hall Ltd, 1984.

TITLE: Business in the 90's: Reducing Time to Action

Author: Steve Jeffery

AUTHOR: For more information contact:

Ella Washington

Hewlett-Packard

19091 Pruneridge Ave. MS 46LK

Cupertino, CA 95014 408-447-1053

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 5211

Logic Programming and Expert System Usage within Hewlett-Packard

Dave Elliott

Hewlett-Packard Company
California Language Lab
19447 Pruneridge Avenue, Building 47L
Cupertino, California 95014
USA

ABSTRACT

Expert systems have now gained wide acceptance in industry for providing knowledge-based solutions to financial, scientific and organizational problems. Within Hewlett-Packard, expert systems and the underlying technology of logic programming are being used in such diverse areas as order processing, product support and CAD/CAM applications. This paper will examine several of these systems with particular emphasis on the rationale for using this technology. Included will be an overview of the tools (shells, languages, etc.) which assist in this approach.

Introduction

Expert system technology is not a new field, but one which has recently begun to realize its potential in enabling significant savings in various processes and products. Simply stated, an expert system is a software program designed to partially or totally replace the need for a human expert in certain diagnostic, prediction, planning and data reduction tasks. It usually consists of a knowledge base (facts or rules), a data base (containing information about a particular situation) and an inferencing system which attempts to apply those rules to the data base. Most rules can be described as 'if-then' statements which attempt to classify raw data into categories or identify courses of action based on the data. Early expert systems concentrated on providing diagnostic capabilities, most notably a system called MYCIN which diagnosed infectious blood diseases.

More recently, major corporations have been using expert system technology for such diverse applications as designing the layout of electric power stations, analyzing mortgage applications, scheduling airport gate usage, diagnosing noise problems in automobiles and estimating construction costs for high-rise buildings.

A myriad of tools are available to assist in the development of expert systems. A 'shell' is essentially an inferencing system without the knowledge base. By responding to a series of questions posed by the shell, the human expert in the domain of interest can build the knowledge base. Among the many excellent shells available on the market are KEE, ART and Nexpert Object. All of these run on HP-UX based workstations and multi-user systems.

Some implementors of expert systems choose to use a programming language such as Lisp or Prolog. This provides greater flexibility and power, but requires more time and programming expertise to develop the systems. At this level, the programmer is directly manipulating the rules and the methods of interpreting them. The techniques form the basis of logic programming. By representing knowledge and rules in terms of predicate logic, the Prolog programmer can have complete control over the operation of the expert system. The great majority of expert systems applications, however, can be implemented quite nicely using a shell.

HP Organizations Involved in Expert Systems

Hewlett-Packard has been involved in this field for a number of years, mainly by manufacturing the computer systems and software often used for expert systems applications. The actual application of this technology within HP has been a more recent development, but there are now several organizations and scores of personnel working directly on reaping the benefits of this field internally.

HP Labs Knowledge Engineering Group (Bristol, UK) - This group, known as KEG, serves as a focal point for expert systems knowledge in HP. They provide consulting services for internal and external users and they communicate news of their activities through internal channels. They have conducted several workshops and technical training sessions and have published papers for AI and expert systems conferences.

Corporate Information Systems (Palo Alto) - As the organization that consults on maintaining and improving the internal flow of information (financial, sales, manufacturing, inventory, etc.) in HP, CIS has been actively engaged in implementing and promoting expert systems and educating the rest of the company on their benefits. They have helped to develop systems for analyzing market research data, tracking the progress of parts being manufactured and evaluating the performance of commodity suppliers. CIS promotes PC-based expert systems, allowing their application in areas that either do not normally use workstations or which require interaction with other PC-based tools.

Knowledge Systems Centre (Bristol, UK) - KSC applies the internal resources and expertise of HP to proposing solutions for and consulting with our European customers.

In addition to these groups, pockets of expertise have formed in numerous HP locations including Roseville Networks Division, Boeblingen Medical Division and Grenoble Networks Division.

The RBEST System

In early 1986, the HP's Disk Memory Division (located in Boise, Idaho) was faced with the challenge of preparing for a very large increase in the number of disk drives to be shipped. One of the potential inhibitors of this growth would be the time required for analysis and correction of test results from the environmental stress testing of each device. Over a period of 24 hours, each drive is subjected to extremes of temperature, humidity and supply voltage beyond the recommended ranges. This can reveal defects in the drive which might not otherwise be found until the unit is in the hands of the user.

The system in place at that time produced a printout for the test results for each drive which ranged from a few lines to 17 pages in length. Containing low-level diagnostic messages, these reports were often difficult and time-consuming to interpret. The test controller was a HP1000 industrial automation computer.

If an error was detected, the interpretation and correction of that error was handled by up to three people. The test operators had a detailed knowledge of the testing procedures and could handle some trivial failures due to incorrect test setup (e.g., a cable not fully plugged in). However, the existence of such seemingly obvious problems could not always be determined from the output of the tests. If the operator was unable to figure out the cause of a problem, it was referred to a debug technician. The technicians had been trained on the product. They had additional diagnostic tests and test equipment available and had known good drives for comparison and component swapping purposes. If the cause or correction of the problem could not be handled by the debug technician, then the debug engineer was called in. The engineers had detailed knowledge of a variety of devices and were expected to handle arbitrarily complex failures. Their primary responsibility, though, was to evaluate the common causes of failure and implement changes in the manufacturing process in order to reduce these failures.

Obviously, the process as outlined could take a great deal of time, both in actually interpreting and correcting the errors, and in communication between the various personnel and generating the required paperwork. Another characteristic of this process was that units with multiple failures sometimes had to go through the process several times before all of the problems were resolved.

Looking back, it can be seen that the situation was one ideally suited to an expert systems approach. The need was for a replication of the knowledge possessed by the technicians and engineers and the ability to apply that knowledge at a much faster rate. An alternative would be to hire and train more technicians and engineers but the cost would have been prohibitive. A consistent knowledge base which could be fine-tuned and enhanced would provide additional accuracy and reliability.

When the need to expand production capacity was determined, a proposal was made to apply expert systems technology to the interpretation of environmental test results. At first, this was intended to be an experiment in the application of expert systems. It would not actually be integrated into the process unless the benefits were clear. The use of the HP1000 as the test controller would be maintained. The expert system would be implemented on an HP9000 Series 300. This was done because of the availability of software tools on the 300. The link is a simple data transfer over a Local Area Network.

One critical decision made during this time was to design and implement the system through iterative prototyping. A series of four distinct prototypes was defined as follows:

1. "*Question and Answer Toy*" - This would demonstrate that a rule-based system was appropriate for this application. The human expert would answer a series of questions about the patterns in the test control system's report and would be given a simple diagnosis.

2. *Batch data* - This would demonstrate the feasibility of a system that could run unattended. The test reports would be fed directly into the expert system.

3. *Integrated system* - Totally automatic data collection, transfer and diagnosis.

4. *Pilot run* - To establish that the initial coverage and accuracy goals for the project had been met.

Each of these prototypes incrementally established the feasibility of the approach and also provided usable tools. The third prototype was able to achieve 75% accuracy in diagnosis over several weeks. The pilot run far exceeded its original goals. In full operation since late 1987, the RBEST system is able to provide a diagnosis for over 99.8% of disk drives tested and its accuracy rate is over 99%. The savings in direct labor costs have been over $1M for a development cost of $150K. The indirect benefits include the contribution that this system has made to the feasibility of the substantial increase in disk drive production at DMD.

The Order Clearing Expert (OCEX)

OCEX analyzes incoming orders for medical equipment at HP's sites in Boeblingen, Germany and Waltham, Massachusetts. Orders are checked for the proper options and configurations. When particular combinations of options are ordered, Manufacturing must be informed. OCEX produces instructions on how to proceed with each order.

Each day's orders are downloaded during the night from the HP3000 in which the main order system resides and checked on the HP9000 running OCEX. Comments are sent back to the HP3000 where they are sent by HPDesk to the originating sales office for action. The processing on the HP9000 takes about ten minutes each night.

OCEX is implemented in Prolog, although the checking rules specific to each product are maintained in a specially-designed language which is tailored to the order-checking task. These rules, which form the bulk of the knowledge in OCEX, cover nearly two hundred separate products, each with many options. They are maintained directly by the users (production engineers, marketing engineers and order processing clerks). It is necessary to update some portion of these rules nearly every day, given the frequency with which new products and options are introduced. There are also frequent introductions of promotional discounts on special option combinations. The knowledge base can be modified to account for these changes. Without the ability for the users to update the system, it would require a large continuing investment of maintenance effort by MIS specialists to keep the rules current.

The savings realized by the development of OCEX includes the direct labor cost and the reduction of the number of incorrect orders sent to production. At

Waltham Medical Division, 70% of the staff previously required for order checking have been moved to more responsible work.

Agatha

This is a joint project between the Colorado IC Division and the Knowledge-Based Programming Department at HP Labs in Bristol. It is currently being tested in the fault diagnosis of HP-PA processor boards. There had been exactly one human expert at analyzing the results of the tests. When responsibility for the manufacture of these boards was transferred to two other divisions, the lack of extensive local experience in carrying out the diagnoses proved to be a bottleneck. *Note: Additional details on this system will be provided during the presentation of this paper.*

Some Other Applications of Expert Systems

This is by no means a complete list, but it does demonstrate the wide range of uses for expert systems at HP.

- helping SE's set up the HP5372A Time and Frequency Analyzer
- trouble-shooting problems encountered by those using the internal asset tracking system
- management of internal communications
- filtering job applications
- assisting Response Center engineers to diagnose software problems
- isolation of LAN problems
- planning the production of parts based on order forecasts
- analyzing problems in manufacturing software
- determining which standards and regulations apply to proposed products

Expert systems technology is also incorporated into several HP products including the HP 4765A electrocardiograph and HPWord (for diagnosing printer problems).

Conclusion

Expert systems are currently the most tangible results of decades of artificial intelligence research. Hewlett-Packard continues to find new areas for the application of this technology. Individuals and groups are actively investigating, promoting and applying these techniques to a variety of tasks with impressive results. Programs that are firmly established in daily use, such as RBEST and OCEX, attest to the feasiblity, maintainabilty and rewards of expert systems.

Suggested Reading

The material in this paper was compiled from a number of internal HP reports, newsletters and conference papers which remain HP confidential. However, there are a number of good introductory texts that are generally available.

Feigenbaum, E., McCorduck, P., and Nii, H.P., *The Rise of the Expert Company* 1988 Times Books, New York.

Hayes-Roth, F., Waterman, D.A., and Lenat, D.B., *Building Expert Systems*
1983 Addison-Wesley, Reading, Massachusetts.

Frenzel, Louis E., *Crash Course in Artificial Intelligence and Expert Systems*
1987 Howard W. Sams & Co., Indianapolis, Indiana.

Lazarev, Gregory L., *Why Prolog? Justifying Logic Programming for Practical Applications*
1989 Prentice Hall, Englewood Cliffs, New Jersey.

Harmon, P. and King, D., *Expert Systems*
1985 John Wiley & Sons, Inc, New York.

Fostering Creativity in a Technological Environment

Kent Hanson
Hewlett-Packard
Corporate Offices
P. O. Box 10301
Palo Alto, CA 94303

Creativity is an elusive term, and therein lies a tale. You will encounter several different definitions, analyses, and suggestions regarding the nature of creativity and ways in which it can be discovered, developed, and managed. Indeed, I will confidently suggest that each of us has a mental image of what constitutes creativity, or what types of things result from the creative process. My perspective on the subject is admittedly subjective, the results of my own personal experience, research, and observation. Nevertheless I maintain that my perspective is also valid and appropriate. Why? Because creativity is an individual quality: a work of art (which most of us would agree is a product of creativity) is characterized by, among other things, the unique individuality of the artist. Without this subjective element, the product is simply a function of the craft, methodology, or algorithm employed to produce it. The term "computer-generated art" seems like a contradiction in terms to me. A computer has neither personality nor personal opinions on any subject. A human being does. I will treat the subject of creativity, therefore, with my own subjectivity.

But what about this audience, and those of you who reasonably expect to get something out of this discussion? This is where subjectivity and personal perspectives must be integrated into objective contexts, common grounds for communication. Without a desire to communicate one's personal perspectives, subjectivity becomes a euphemism for self-centered ego-gratification. My objective is to bring my perspectives into the arena of objective discussion, to use terminology and concepts that I trust will be meaningful, understandable, and fruitful for my audience: I do not intend to be abstruse or vague.

This combination of individual subjectivity and its immersion in the objective world through communication is to me exactly what creativity is all about. I believe that the creative process involves the application of indeterminant, internal, psychological, subjective, unconscious dynamics into the determinant, external, practical, objective world. The creative process, therefore, will be the guiding process of this paper and its presentation, so that we can experience the very subject we are discussing as it is being discussed--kind of a "self-fulfilling prophecy".

I will begin by asserting that there is no such thing as creativity--or that there might as well not be--and I will then set out to demonstrate that this assertion is incorrect. I will maintain that the creative process cannot be managed, yet I will itemize several methods and techniques for managing creativity. These seeming paradoxes are integral to the creative process, the ability and willingness simultaneously to take nothing for granted while being grounded by a clear and unshakable objective--in this case a personalized but appropriate means of managing the creative individual in the technological environment.

CREATIVITY AND TECHNOLOGY

Because most of this audience works in a technological environment, I feel that it is necessary briefly to discuss the technological milieu as it relates to the subject of creativity. Having done that, however, we should move on with the understanding that creativity can be developed, nurtured, and exercised in any context, technological or otherwise. Technology, as defined by Charles Susskind in his book *Understanding Technology*, is "...man's efforts to satisfy his material wants by working on physical objects." More specifically for our purposes, I would describe the technological environment as being characterized by the presence and application of automated tools and processes to facilitate problem-solving and communication. The technological tools with which we are most familiar relate to computerization. Many of us have encountered a general mistrust of computers at some time in our lives. Sometimes this mistrust is expressed in a "Brave New World" type of literature which depicts a society in which computers and automation have completely "taken over"; at other times the mistrust is embodied in more "home-spun" attitudes by people who prefer to do things "the good old way," and who see the computer as an invasion of their personal lifestyle. We who use computers daily may see them at times as occasionally frustrating--yet useful--tools of our trade. Whatever the perspective, I would say that there is a fundamental dichotomy between technology--computers, automation-- and the "human" factor.

Technology, when used as a problem-solving tool, can be very positive, beneficial and productive. The potential negative element arises when technology isolates individuals from the people with whom they are working in a cooperative setting. Employees whose jobs involve the regular use of computers may find themselves at times distanced from the social aspect of interpersonal dynamics. The most fundamental form of this potenial isolation is the use of technology in communication: electronic mail necessitates the communication of ideas in a written form, without the human elements of vocal intonation and body language, gesticulation; voice mail provides the vocal element but does not allow for an integral part of communication between people--dialogue. The time between statement and response is protracted through the use of technology in communication. This is a "red flag" for managers who wish to promote individual creativity and group synergy.

As our offices become increasingly automated, as we become increasingly focused on our individual projects (symbolized by the "personal computer"), we should at least be aware of the sometimes subtle ways in which technology may have a negative influence on our "humanity." Because creativity is a specifically human phenomenon, there is always a potential danger that working with impersonal technological tools will thwart the development of the human qualities which facilitate creativity. Nevertheless, I believe that the technological and the humanistic are not mutually exclusive: creativity on its own is nothing until it is expressed in an understandable context with definite parameters, and the technological environment is just such a context. The objective structures of technology can provide--as much as music or paint--a vehicle for human expression, and can actually facilitate personal creativity.

The fine arts are no longer considered to be the exclusive domain of creativity. The study of creativity by scientists and business people is accelerating, if only because there is a growing awareness of the value of creativity in personal development, innovation and social progress. Scientific studies have gone so far as to attempt to pin down the neurological processes involved in creativity. The application of creativity to business issues is a field whose time seems to have come. Several reports, books, and courses currently exist which focus on creativity in the business world. The archetype of this trend is the course offered at Stanford's Graduate School of Business, entitled "Creativity in Business," and the book of the same name which was published in 1986. The concept of business as an art form is gradually gaining acceptance.

Now for a word or two regarding my "credentials." I have been employed at Hewlett-Packard's Corporate Headquarters for about twelve years, in various positions at the Corporate Computing Center. After four years as an individual contributor in Operations, Systems Support, Applications and Systems Programming, I was made a manager. I currently manage one Systems and two Applications programming groups. While I have been involved in the technological environment in these two capacities (as individual contributor and as manager) I have also been highly involved in several artistic and academic pursuits such as music, literature, philosophy, the humanities, and the social sciences. I also taught Computer Science at a local college for about five years.

The impetus for this paper is the creative fulfillment I have experienced through the integration of my academic and artistic ("humanistic") endeavors into my jobs in the technological environment. The paper's content has emerged through my survey of available literature, from dialogue and group discussion with employees who report to me, and from my own personal experiences. Moreover, I have found through my own application of the principles discussed in this paper that fostering creativity in one's employees can be very rewarding: I believe that the people who report to me are successful, motivated, productive, and happy in their jobs. Nothing generates credibility so much as results.

The manager's awareness of and sensitivity to the needs of the creative individual, and the manager's willingness to foster that creativity by doing his or her best to produce an environment conducive to creativity, will result in a win/win/win situation for the employee, the manager, and the company. This has been my experience and, from all accounts, the experience of those individuals who report to me.

WHY CREATIVITY?

I believe that the first step in any creative endeavor is an explicit or implicit question which gets to the heart of the matter, a question such as "Why am I doing this at all?" or "What does this matter?" The fundamental question we must ask ourselves before entering into our discussion on managing for creativity, therefore, is "Why should we care about creativity at all?" In order to answer this question, we must know our answer to a still more fundamental question, namely, "What is important to me?", in this case "What is important to me as a manager?" and, more specifically, "What is important to me as a human being?" I will give you my answers to these questions, and I will then tell you why I believe that, based upon those answers, there are good reasons to focus on the subject of creativity.

It is important to me as a human being that I am happy and secure, that I have my needs met and that, having my needs met, I operate effectively in and contribute positively to my physical, domestic, and occupational environments. The accomplishment of those objectives also involves the fulfillment of my role as a manager. What is important to me as a manager is that I lead, guide, and develop my employees in the best ways I can, which turns out to be the ways I would most like to be led, guided, and developed as an employee. It also is important to me that I satisfy the expectations of my managers and the company which employs me and provides me with the fulfillment of many of my most fundamental needs. I perceive those expectations to be the hiring of quality people, and the development and guidance of those people into productive workers.

Objectives may vary slightly between companies, but I think it's safe to state that a primary objective of any company is to stay in business, to be healthy and growing (which means making a profit) by providing quality services and products to its customers, thereby generating trust and confidence among those customers. It turns out that the needs of the company are analogous to the needs of the individual contributor. If employees are satisfied and growing in their jobs, their productivity and the quality of their work will contribute to the quality of the product received by the customer. This is what any company desires. The manager, therefore, stands midway between the employee and the company, harmonizing the objectives of each. Of course all bets are off if the manager does not sincerely wish to be the best manager he or she can be, if he or she does not see the fulfillment of his or her needs/objectives as being integrally linked with the needs of his or her employees and the needs of his or her company. For the purposes of this discussion, however, I will assume that the managers in this audience share my concern for fufilling the managerial role in the best way possible for themselves, the employees they manage, and the company for which they work.

Having established the manager's desire to fulfill his or her role as well as possible, and having associated this desire with the needs of the employees and of the company, I suggest that there are several ways in which creativity can be associated favorably with the objectives of the individual contributor, the manager, and the company. For the individual contributor, the practice of creativity contributes to job satisfaction by giving the employee a sense of unique contribution, something which is decidedly his or her own which also fits well with established departmental and company objectives. Although my surveys indicate that job satisfaction is more important to my employees than "tangible" rewards, nevertheless those rewards are also important, and productive individual contributions should regularly translate to tangible rewards such as recognition, promotion, and raises. Job satisfaction and tangible rewards will stimulate and increase motivation, the employee's willingness and enthusiasm towards the accomplishment of his or her tasks and the fulfillment of position objectives, which will result in increased productivity. Job satisfaction and motivation are indispensable components of morale: high morale in individuals will contribute to more effective interpersonal relationships at work and to productive group dynamics and teamwork. The individual contributor's exercise of creativity also stimulates the development of a quality which is very valuable in businesses making use of modern technology--adaptability. The employee, by being an agent for change, becomes more open to change, more flexible in the midst of an increasingly dynamic environment, better able to accomodate and to assimilate new technologies and business needs.

A manager who promotes creativity in his or her employees will reap the benefits of having happy, self-actualized, motivated, productive employees. His or her employees will be less prone to the natural estrangement between the manager (as a symbol of authority) and the individual contributor, more loyal to the manager and consequently more open to the manager's leadership and guidance. This loyalty will also

Computer Museum

result in increased harmony between employees, because the manager has facilitated job-satisfaction in individuals, who, having their job-satisfaction needs met, can then function as a team without the need for destructive competition or strife. The group will then be an effective, healthy team of satisfied employees, who can put into operation the principle of synergy, in which the accomplishments of the whole are even greater than the accomplishments of the individuals comprising the team.

When employees are satisfied with their jobs, it follows that continuity will be produced: employees who are happy will stay in their positions longer, giving depth to the group and providing valuable sustained experience levels for developing increased heights of expertise and maturity. Employees who trust their manager will also seek to support him or her as a human being, and will not subconsciously attempt to undermine the manager's efforts through underachievement or passive resistance to new ideas and progress. The manager will consequently reap the benefits of his or her employees' productivity through his or her own recognition, promotion, and raises as he or she acquires the reputation for getting results. The manager will also share in the intangible but very important reward of job satisfaction.

Employees whose managers foster the exercise of individual creativity will contribute to company objectives through their loyalty to the manager and to the company, because fundamental needs for self-expression and growth are being met. Innovation is essential as a company seeks to grow and to apply new technologies and solutions to existing and expanding customer business requirements. The creative employee is both innovative and productive, developing novel and efficient solutions to problems, improving processes and developing products which fit customer needs. Satisfied customers will be loyal customers, and their confidence in a company's ability to adapt to changing requirements in innovative and progressive ways will translate to the company's growth through the bottom line: sales.

The needs of the company are, in the final analysis, based upon the individual needs of the employee--the structure does not stand without the foundation. The primary need of the employee is job satisfaction, and I have found that if an employee can practice creativity, he or she will have increased job satisfaction. Many--perhaps too many--employees are job-hopping as they search for ways to express their individuality in a context they hope will be conducive to their personal creativity. It may be that some jobs are more appropriate than others for the facilitation of an individual's creative expression. This is why managers should be very careful to hire individuals whose skills and career objectives match as well as possible the position for which they are hiring. I believe, however, that it is less a matter of matching an individual to an "ideal job" than of providing that individual with the best environment for them to flourish in the position in which they find themselves. Creativity can be applied in any context and, once an individual is employed, it is the manager's responsibility to foster individual creativity within the parameters of the position's objectives.

WHAT IS CREATIVITY?

Now that we have established something of the relationship between individual creativity and the needs of the company, we can venture further into the details of what actually constitutes creativity. Again, creativity is hard to pin down or to define in a sweeping statement. It is analogous to the wind, which is known more by its effects than by its essence. We can probably come closer to a "definition" of creativity by not seeking to define it at all, but rather by identifying some of its effects.

We will begin with a brief discussion regarding the so-called "creative process," the process through which creativity is exercised in a specific context. In the technological environment we can identify that context generally as "problem-solving," which implies the use of specific tools and skills to arrive at a solution to a specific problem. Any process should result in a product, so we will next discuss the result of the creative process, the "creative product." Because creativity may be applied to any situation, creative products will be as diverse as the situations in which they emerge. For this reason we will not seek to define the products themselves but rather to describe some of the characteristic features of these products, the "fingerprints" which demonstrate the presence of creativity in action. Having identified these features, we will then move on to a discussion of the individual qualities in a person which should be present in order to successfully exercise creativity and to generate a truly creative product. Finally we will discuss ways in which the manager can nurture those individual qualities in his or her employees.

The Creative Process

The "creative process" (like any process) is a continuum, the parameters of which are a problem to solve or an objective to be achieved, and a product which solves the problem or achieves the objective. Once again we must begin with fundamentals, in this case, the prerequisites which must exist before an individual embarks upon the exercise of the creative process. In problem-solving these prerequisites are first a clearly-defined objective and next the knowledge, experience, and technical skills at the individual's disposal to apply to the solution of the problem. These raw materials, however, are meaningless without an even more fundamental component, namely, an attitude of conscientious determination to solve the problem. Without this implicit attitude, discussion of any problem-solving process--creative or otherwise--is meaningless.

With these fundamental prerequisites established, the first step in the creative process is *focus*. This involves intense concentration, an immersion into and an internalization of the problem or the task at hand. This phase should be used primarily to understand the problem as comprehensively as possible, to attempt to see it from as many different perspectives as possible, and also to identify and mentally test possible solutions or paths towards the ultimate solution to the problem. During this phase of the creative process the individual should refrain from jumping to quick or immediate solutions, and should hold in reserve the tools at his or her disposal while seeking to understand the situation fully.

If the creative process is functioning well, the individual will consistently encounter obstacles to the problem's solution, "dead ends." If this is not happening naturally, the individual can actually facilitate this frustration by not accepting the most obvious solutions, by "holding out" for an unexpected, novel and appropriate solution, i.e. a "creative product." This may seem self-defeating, but if the individual has the luxuries of time and the freedom to allow the creative process to run its course, he or she will balance the unwillingness to accept any solution with an awareness of when it is appropriate (or necessary) to make a decision.

It is essential in the creative process to balance this intense focus with a period of total removal from the problem. This withdrawal is sometimes referred to as an "incubation period," during which the subconscious takes over. This is the most vague and imprecise--yet the most essential--part of the creative process. The analytical gives way to the intuitive, or, if you prefer, the left-brain gives way to the right-brain. The result is a kind of a picture, or "visualization" of the problem, and it is important that this part of the creative process be given the time and the freedom to occur without pressure. Artists consistently speak of their work emerging in a situation in which they had given up focusing on the work itself and were engaged rather in some pleasurable or recreational activity, totally removed from their project. Relaxation seems to be indispensable to this phase, as psychologists have noted in studies which indicate that sleep and dreaming often bring to light unexpected and appropriate solutions to problems. Although we do not understand fully just what is taking place during the "incubation period," there is enough evidence to suggest that it is essential to the creative process.

The creative process, therefore, involves a balance between focus and withdrawal, between hard work and rest. This balance can be cyclical, that is, it can be repeated as often as is feasible or necessary. What is happening is that, through each cycle, new perspectives on the problem are being added, and new insights are emerging. This process is sober and measured, and cannot be forced. Unfortunately, we have severe pressures which impel us to accelerate the process, and we often simply do not have the luxury of applying these "techniques" to every problem or for allowing the process to take place in its own time. Because of our constraints, the elements of responsibility and priority should be included as factors in the creative process: depending upon the criticality of the problem, the individual should be able to control to some extent the pace of the cycles of focus and withdrawal. It should be noted that as the individual continues the exercise of the creative process, it should become increasingly easier for him or for her to enter into it and to control its pace.

The hopeful outcome of this process, of this balance between focus and withdrawal, is the creative product, the "light bulb" being turned on. This aspect of the creative process has been referred to as "insight," "illumination," even "revelation," and it involves a certain confident "knowing" that the solution has been reached. The creative product is usually elegant in its simplicity, totally appropriate and comprehensive as a solution to the problem at hand. At that point it remains only to verify the product's soundness through the test of implementation.

The Creative Product

Now that we have discussed the creative process, let's examine the result of that process, the "creative product". As is becoming clear throughout this paper, it is necessary first to establish a certain objective criterion before we move on to the discussion of details relating to it. Just as we are assuming that the individual will bring to the creative process a conscientious determination to solve the problem, so also we should assume for our purposes that there must be real value to the creative product. Creativity can be either constructive or destructive; we will seek to identify those aspects creativity which will be useful and helpful for the objectives of all involved, and we will use as an implicit criterion throughout the discussion that the creative product will be constructive.

Teresa Amabile has written several books on the subject of creativity, and her organization, the Center for Creative Leadership, has undertaken several "scientific" studies of creativity. Ms. Amabile has defined the creative product as being "novel and appropriate, useful, correct, or valuable..." (quoted in Ray and Myers, *Creativity In Business*, p. 4). I can live with this definition long enough to use it as a conventient objective framework within which we can discuss the characteristics of the creative product. We have already established that the creative product will be constructive (useful, correct, valuable); I will focus primarily on the creative product's novelty and appropriateness.

First of all, then, the creative product is something *new*. This does not mean that nothing like the product has ever been seen before, but rather that the product represents a unique solution to a certain problem or situation. The Pascal language is certainly recognizable as a programming language, but its invention represented new approaches to the solution of programming problems which had not been adequately addressed by prior languages. Although the creative product's source is unconscious inspiration, it does not just come in "off the wall"; the individual exercising the creative process in a particular situation brings to the problem accumulated experience and knowledge appropriated from the work (creative or otherwise) of others, groundwork laid in preparation for the new idea. Shakespeare's works would not have been possible without the Phoenicians' development of the alphabet. The creative product may be unconventional and is by definition original.

The whole idea of "creating" something is the idea of bringing something new into existence. For this reason, every time you doodle during a meeting you are being to some extent "creative." Whether or not this is constructive creativity is a subjective evaluation.

Secondly, the product of constuctive creativity is something *appropriate*. That is, the product eminently suits a certain need, solves a particular problem, applies well to a specific situation. It will be helpful, useful, and functional for that situation, even though, by its novelty, it may be quite unusual in terms of familiar, traditional reference points. Its appropriateness will be tested by its functionality, and also by others' ability to understand it. A creative product will have little or no value if it cannot be understood. Another test of the product's appropriateness is its ability to be integrated into the larger context. For our purposes this means that the solution developed by the individual in response to a specific problem relating to his or her position objectives will ultimately have some real value for the objectives of the company.

THE "CREATIVE INDIVIDUAL"

Having briefly described the primary features of the creative product (novelty and appropriateness), we can now proceed to a discussion of the qualities in an individual which should be present to facilitate the practice of the creative process, and the development of a creative product. Once again before we proceed we must ask another fundamental question: is there really such a thing as a "creative individual" or is the potential for creativity latent in everyone? Although it may appear that certain people seem to practice creativity naturally, and that others may seem to be decidedly non-creative, I have found that a creative individual is made, not born, that given the raw materials of intelligence, skills, experience, and a problem to solve or a task to accomplish, any person should be able to undertake and practice the creative process.

Novelty

In order to create something new, a person must be open to the possibility of novelty. This means that they should be ready for anything, including something they have never encountered before. They should have a childlike, almost "naive" attitude towards the possibilities for solutions to their problem or the task at hand. This is much easier said than done, especially for us sophisticated adults who, whether or not we admit it, have a plethora of preconceptions and a fairly rigid idea of the way things "should" be done. Nevertheless it should be possible, with practice, to put our preconceptions aside and to approach each problem, each task, each objective, as though it is a totally new situation (which, in the final analysis, it is--you never step into the same river twice). How do we do this without denying our fundamental understanding of the problem's context, or its technical components? After all, the wherewithal to solve a problem ultimately comes from our own technical experience and skills. But the trick in following the creative path towards applying those skills involves a kind of "self-deception," in which we assume for a time the innocence and sense of wonder a child has. We "pretend" that we know less about the problem than we may actually know. We are prepared to look at it in a fresh, unbiased way. This is the attitude and practice of *openness*.

Because we have opened ourselves to unlimited possibilities and are treating each situation as a fresh, new challenge, we can contribute to the accomplishment of something new by applying an attitude of *experimentation*. Exercising creativity involves the willingness to try possible solutions to our problem, or to the problem's constituent parts. Curiosity is a big factor in this phase of creativity, the curiosity to wonder if something will work, and to wonder why it did or didn't work after we have tried our solution. I am reminded of Thomas Edison's famous response to the question of why he continued to persist in his search for the incandescent bulb, after having failed more than nine thousand times: "I haven't even failed once; nine thousand times I've learned what doesn't work." (Quoted in Ray and Myers, p. 54).

There is an element of playfulness in this attitude: "I wonder how this will work", or "Let's try this." Now, I expect a reaction of at least caution and perhaps even suspicion towards the idea that an employee will be trying out endless, playful experimentation--not exactly a picture of productivity. But you will recall that we have established certain ground rules for the creative process: that the person understands their objective, and that they are sincerely willing to find a path towards meeting that objective fully. The manager does have a responsibility to recognize when experimentation is floundering or straying from the "critical path," which means veering too far away from an understanding of the objective. Also, time and priority constraints will necessarily limit the potentially endless experimentation process--there comes a time when you've got to go with what will work best, it's not as though you have a government grant to support you while you fully work out the scientific method. Nevertheless, experimentation is integral to the creative process, and without it it is unlikely that anything really new will come about. Something new, by definition, has not been seen or tried yet, and before it can be put forth as a solution, it must be suggested as an alternative, necessitating trial, experimentation.

The experimentation aspect of the creative process, by the way, need not take place in the arena of the actual problem; that is, the person should have the ability to visualize the outcome of his or her experimental suggestion, kind of a "mental experimentation." Einstein did not work out his theories of relativity in a laboratory. This and other skills which comprise the creative process develop only with practice.

Openness and experimentation provide the framework and the method for progressing towards something new. Beyond these, the individual who is breaking new ground, blazing new trails, swimming in uncharted waters, must have certain personal characteristics. One of these is *independence*. The innovator will be open to possibilities that will certainly vary from traditional or standardized ways of accomplishing tasks, and for this reason he or she will be in some senses on their own, willing to dance to the beat of a different drummer. Their colleagues may look askance at some of their experimentation, some of their new ideas, but their manager should not; the manager should instead encourage this independence, for it is essential if progress and innovation are to take place. In this sense each person can be a leader. This may threaten some managers, who feel they need to have control over their employees, that they must have all the ideas. But it is far better to encourage each individual to have faith in their own personal exercise of the creative process, to be independent and to make independent judgments, while also recognizing the importance of participating on the team.

This independence will put the employee on the line, will take him or her out of the security of falling into step with group, the department, or even the company. For this reason there is definitely some personal risk involved; the creative individual therefore needs *courage* to remain independent, to pursue the path of experimentation in the freedom of openness in the face of traditional taboos and possible isolation. The innovator is frequently considered to be a nonconformist; the stereotype of the disheveled, eccentric, absent-minded but brilliant engineer is one we all know well. These "types" are either tolerated or ostracized, depending upon the situation. How many of us would consider actually *encouraging* eccentricity? But to me "eccentricity" means that a person's individuality is showing, that their habits or perspectives do not necessarily fall into step with institutionalized patterns, that they do not "fit the mold." This is exactly what I, as a manager want--individuality, independence, and the courage to be one's self, tempered and guided in the work context by a conscientious and sincere pursuit of the objectives at hand. So long as I am confident that my employee is doing his or her best faithfully to accomplish their objectives and to solve the problems for which they have been given responsibility, I can certainly accomodate their individual eccentricities.

With fundamental skills and experience, an open and experimental attitude, and the independence and courage to move forward on one's personal creative quest, accepting the risks involved, the employee has everything necessary to produce something new. These components, however, mean nothing on their own: you can toss all the words in the dictionary into the air and you'll probably never come up with a truly meaningful sentence. What provides the meaning, what brings all the "raw materials" of the creative process together? What produces that moment of insight, revelation, "eureka" which gives birth the creative product? We are again speaking of that intangible quality, the quality of "intuition." This quality is integrally wrapped up in an individual's psyche, and consequently can be nurtured primarily through the manager's attention to the employee's psychological needs. You cannot control intuition, in fact the harder you try to force it (with schedules, stress, pressure, methodology), the further away you will probably send it. You must be content to trust your own nurturing of the employee's creative development: each new creative approach to problem-solving is a potential for failure, and you cannot really control the outcome. This ambiguity tends to make us managers uneasy so we often resort to procedures, methodologies, standards, trying to reduce our tasks to formulae (algorithms) which will produce guaranteed outcomes. But true innovation requires a "heuristic" approach, an independent quest for discovery in which all the past experiences, current investigations, and the exercise of the creative process come together through the operation of the unconscious to produce a creative product. It is up to each manager to weigh the risks involved in really trying to foster creativity in lieu of simply cranking out the work through mechanical, tried-and-true methodologies. Perhaps some healthy experimentation on your own will give you more faith in this area. I know from my own experience that it pays off.

Fostering Creativity - Novelty

The manager can foster and establish an environment conducive to the "new" aspect of creativity by welcoming, encouraging and supporting new ideas. Any natural reluctance to new ideas should be balanced by an understanding that without novelty there will be little creativity or progress. It should not matter if the ideas are in the formative stages: the manager can encourage novelty by accepting the ideas where they are at, even if not yet fully developed, and by trusting the employee's exercise of the creative process to enlarge upon and to expand them. A new idea is a fragile thing, and the manager has the power to deflate or to bolster an employee's confidence and independence when presented with a new idea. The manager should be comfortable with each employee's individuality, for it is from this personal stamp that real innovation will occur. No two people see things in exactly the same way, and for the sake of innovation it is good that they don't. Each person's unique perspective is the potential for something really new.

The manager can strongly contribute to the environment where novelty is encouraged by being his own person as well, by submitting his or her own unique perspectives on issues which arise in the group. But there are even more creative ways of doing this: just as the creative process may be applied towards a given task or objective, the manager can apply the creative process to the group he or she manages by being open to and by encouraging exposure to unusual stimuli. Why not contribute to the environment of novelty by a kind of adult "show and tell" period, where each person brings to work something unique from their own life which they describe, explain, "share" with their colleagues? Just a suggestion.

Specialization can be an obstacle to novelty. If the manager wishes to encourage the employee to be open to new approaches or ideas, he or she should provide the employee with a variety of projects which will increase the perspectives and skills at the employee's disposal. The manager should also take care to see that each employee is regularly provided with new challenges, hopefully different in some ways from prior projects: the more diverse and unrelated subsequent projects are, the broader an individual's experience will become, and the better able the employee will be to accomodate new situations, and to avoid falling into the rut of "the way things are done". Some sort of mechanism for considering new ideas should also be provided by the manager (perhaps a "suggestion box"). The "What If?" concept is quite appropriate here. The manager can ensure that his or her commitment to novelty is more than simple "lip service" by actually *implementing* new ideas into the mainstream of group or department processes.

As mentioned above, novelty will bring with it a certain risk, the risk of change, of implicit destabilization, which people simply do not like. As we also mentioned, the creative employee will run the risk of being outside the mainstream. The manager should be willing to share that risk. He or she should be ready to defend each employee's individuality, and should understand the employee's new ideas to the extent that he or she can accurately verbalize and defend them to other managers and persons outside his immediate department, as well as to the colleagues of the employee. If the employee feels sheltered from criticism and prejudgment, he or she will feel much more secure and courageous about the risk of novelty, and the pursuit of their independence.

Instead of carrying predispositions and prejudices against new ideas, the manager should take on faith that the employee's novel suggestion has merit. This does not mean that the manager should accept without questioning every new idea that comes up: that would only communicate a message of non-interest, of not really caring. Instead the manager can demonstrate his or her interest by dialogue, encouraging novelty with constructive questioning, providing his or her own unique perspective, again, without preconceptions or predispositions. The questioning attitude should be part of a positive process in which it is understood that both the manager and the employee are sincerely and responsibly seeking out fresh, new, and adequate solutions.

Outside the parameters of the task at hand, the manager can encourage novelty by such things as off-site meetings to places or experiences unrelated to the department's charter. I personally like the idea of "field trips" to an art museum, to a concert, to a play. This type of activity will not only contribute to the value of varying perspectives, but will also contribute to the personal development of each individual through new experiences, and to the group's bonding through socialization.

The manager should be consistent but should not be predictable, should demonstrate his or her own flexibility, adaptability and personal growth along with the growth of his or her employees. The manager should be able to play with ideas right along with the employee, and should have the mental agility to follow the employee's train of thought. This means that the manager should be able to shift gears to adapt to the thought processes of several different people, no easy feat but certainly worth the effort.

Finally, there should be a balance between fostering individuality and in harmonizing the team. As it is in music, so it can be in the work group: each note in a chord is different, unique, totally individual, yet together they create something new, synergetic, greater than the sum of the parts.

Appropriateness

The second main feature of the creative product is *appropriateness*, and for an individual to create a new product that is also appropriate to the environment in which he or she is operating, certain prerequisite components (once again) must be present. Again, it is essential that the individual has a controlling attitude of conscientious awareness of the specific objective in order to move towards an appropriate outcome. This is the "microcosmic" aspect of appropriateness, the specific task at hand. Something new, from "off the wall" only has value in constructive creativity if it works for the situation at hand. Here we see the "serious" counterpart to the open, playful elements which contribute to the environment of novelty. (You knew you'd have to eat your vegetables if you wanted dessert). In this case the employee must have a realistic attitude towards the objective. This includes an awareness of his or her own limitations: his or her expertise level, how much time is available to accomplish the task, available external resources. These awarenesses provide a good balance, or "check" to the freedom and openness which generate novelty.

A very important test of a creative product's appropriateness is its assimilation into the global context. This is the "macrocosmic" aspect of a creative product's appropriateness. The employee should understand not only the fine details of the problem at hand, but also how the problem and its eventual solution fit into the "big picture" of departmental and company objectives. This "integrateability" is vital to constructive creativity, and I will suggest that the creative product will often solve more than one problem and may apply to more than one context. The synthesis of several distinct elements into a meaningful whole is one of the creative product's most characteristic features. In order to effect the appropriate element of the creative product, the employee should be able to assimilate and to correlate a wide variety of experiences into the problem at hand. Arriving at an appropriate solution to a problem does not necessarily require the application of knowledge specifically related to that problem. Indeed, the ability to correlate knowledge from several areas into a unified, local focus is an integral part of the creative process. This skill is sometimes referred to as "convergent thinking."

Obviously, in order to simultaneously hold as much knowledge, information, and experience as possible in reserve for application to a particular problem, the individual must have a well-developed *memory*. The manager should do everything possible to provide the employee with the opportunity to develop his or her memory skills (mnemonics). I cannot emphasize this enough: a fully-functional memory will greatly facilitate the available reference points--relevant or not--for correlation into a problem's local context.

Correlating seemingly unrelated experiences or situations involves extracting principles which apply to both, underlying commonalities which may not be immediately obvious. I suggest that the employee and the manager can practice developing this skill by thinking and speaking analogically ("This program is like a ship at sea...") or metaphorically ("Our group is a spectrum comprising unique shades of color...") Interestingly enough, *humor* promotes this aspect of appropriateness, this "integrateability." This is true because humor often involves the perception of incongruities, the unfamiliar in a familiar context. The more humor the manager can provide, therefore, the better for an employee's practice of correlating unrelated ideas. Has anyone considered bringing a stand-up comic on-site? The employee does not operate in a vacuum; he or she should have the ability to move easily from the detail to the global, from the trees to the forest and back again. Once again we see that creativity is a matter of balances.

Fostering Creativity - Appropriateness

The manager can foster an employee's focus on appropriateness first of all by establishing clear objectives, and by making very certain that the employee is aware of the time allowed for the accomplishment of those objectives. The manager should contribute to the employee's ability to appropriately solve a problem by facilitating training for the development of appropriate skills. The educational factor cannot be emphasized strongly enough. The manager should ensure that each employee has the fundamental skills and tools to give them a "head start" in applying the creative process to the problems they are given. But beyond this, I believe it is the manager's responsibility to ensure, so much as is possible, that each employee is given the opportunity to acquire knowledge in areas *outside* those areas directly relating to the job. So much as time and opportunity allow, I would strongly suggest bringing in guest lecturers for seminars on everything from classical literature to stress management to creativity in business. I myself intend to offer a course in music to the individuals who report to me.

MANAGING FOR CREATIVITY - PSYCHOLOGICAL FREEDOM AND SECURITY

Creativity, elusive term that it is, relates closely to a person's psychological life. Managing for creativity, therefore, boils down to fostering an environment of psychological freedom and security for the individual. I would like now to mention some of what I consider to be the most important qualities in a manager which will contribute to this psychological freedom and security, and which will ultimately foster the growth of creativity in his or her employees.

First and foremost is respect for the individual, understanding where they are coming from, accepting them for who they are, appreciating their strengths, not condemning their weaknesses. The manager cannot adequately do this unless he or she feels confident in himself or herself. An insecure manager may be threatened by the accomplishments of his or her employees, and may appropriate those accomplishments as his or her own rather than giving credit where credit is due. He or she may become defensive about an employee's individuality and may attempt to make the employee conform to some kind of "template" or role which thwarts the individual's free expression. The manager who seeks control of group policy, who is autocratic, will stifle the truly innovative ideas that can provide progress to the group and to the department. A more positive analogy would be towards representative democracy, where all employees play a part in the decision-making process but have granted the power for the final decision to the manager, whom they trust. The manager in return should trust the employees as individuals to contribute value, should have faith in their ability to contribute.

Employees who feel as though they are part of the decision-making process will feel good about responsibility. Time after time I have seen individuals who were considered "under-achievers" rise to the occasion when given the responsibility for truly important and critical decisions and projects. Delegation in this case turns out to be a relief for the manager and a vehicle for growth in the employee.

The manager should expect responsible, conscientious work from his employees. He or she may be disappointed if and when the employees take unfair advantage of their autonomy and freedom--this is always a risk when a manager is truly attempting to contribute to the personal growth and independence of his or her employees. But where the manager has taken a sincere interest in the employee as a person, a personal relationship between the employee and the manager will have developed, and the manager should feel free to confront irresponsible behavior, to express his or her disappointment. In an environment of trust and honest relating, the employee should have enough integrity to be sensitive to the manager as a person and to seek to rectify the damage that has been done to the relationship. This is a much more positive approach to the problem of unacceptable performance than threat or retaliation. Some of you may look askance at this personal interrelationship between manager and employee. Does it seem inappropriate in the business context? I think not. The manager should not become necessarily a totally forgiving "buddy" to his or employees, but on the other hand I believe he or she should be personally involved with each. Artificial loyalty based on hierarchical relationships will not be nearly so effective as the loyalty that comes from friendship. I make certain to meet with each of my employees in a one-on-one context regularly, to discuss not only work issues, but also personal issues of my own and of their own, philosophies of life, anything at all. There is certainly a risk in this approach, but without the risk of vulnerability there can be no trust.

The manager should not be so wrapped up in his or her "role" that he or she projects some kind of image that covers over his or her existence as a person. Modesty plays an important part in this. The manager should also be available to his or her employees. Each employee will be experiencing highs and lows at various times, and the manager should be adaptable enough to accomodate their needs, especially their most important needs: reinforcement and supportiveness. The manager should be consistent: if the employee knows what to expect from the manager, he or she will be trusting and open. The manager should naturally treat each employee as an equal with himself or herself as well as with the other employees in the group, sincerely recognizing each employee's unique contribution to the team and avoiding the natural tendency for "sibling rivalry." The analogies between managing and parenting are obvious, but the manager should never be condescending or treat his or her employees like children.

The manager should act as a buffer zone between the employee and the inevitable political issues which arise in any organization. He or she should mitigate red tape, frustrating formalization and superfluous interruptions to the employee's continuum of problem-solving, and should shoulder these burdens on behalf of the employee. The manager should be sensitive to where each employee is coming from individually, socially, and culturally.

These are not easy tasks for any one individual, it takes a lot of self-sacrifice to be a good manager. But every organization needs good managers because every organization needs satisfied, cooperative, productive employees.

A manager can study organizational behavior or psychology as much as he or she wishes, can take course after course in management techniques, but in my opinion there is no surer way of ensuring the employee's security than to practice compassion, caring, treating the employee as you yourself would prefer to be treated--true role-modeling. But don't write it down as a "methodology". The creative process is internal, and the manager's ability to nurture it must come from within as well. The bottom line is sincerity: people can sense insincere phraseology or management "techniques" that go through the motions of formulae such as "positive feedback." The methodology I am advocating is a "non-methodology," it is the result of a creative process which I suggest involves an intense focus on the principles we have been discussing, followed by a period of withdrawal and "incubation." Unconsciously-held beliefs and attitudes are much more potent than methodologies we practice. The good news is that the more we practice certain techniques and attitudes, the more they will become a part of our unconsciously-held attitudes. So practice away! While you are sincerely developing an internalized attitude which fosters creativity in your employees, they will probably accomodate your lapses into hesitant attempts to "try" something which cannot be forced. When the employee feels as though the manager really does care about their growth, their ideas, they will be loyal and secure knowing they have real value as an individual.

By now it should be clear that the creative process cannot be managed directly, it can only be facilitated through the manager's attention to things which will contribute to an individual's personal development. There can be no question of creativity at all unless the employee is at least somewhat secure and happy in his or her job, unless the job is more than just a daily drudgery to which the employee brings a modicum of motivation and for which application of energy is a real strain. And there is no guarantee that, once the manager has provided a healthy environment for individual development, creativity will naturally ensue. But it is much more certain that where psychological inhibitors exist, creativity will be thwarted. Most employees will put in their eight hours, will apply their skills to the accomplishment of their position objectives: their motivation may be primarily to keep their job, to advance their career, to support themselves. There is really nothing in the unwritten rules of management that says you should actually try to make better, more fulfilled people out of your employees. Each manager must weigh the risks involved in venturing into the non-concrete world of the unconscious, the intuitive, the psychological, the creative. I can only speak for myself and for the employees I manage: I know that being able to practice creativity at work is a rewarding experience, that it provides personal growth and fulfillment, and that it adds an element of job-satisfaction and yes, happiness to the daily routine. I also know that when my employees are happy, I am happy.

SELECT BIBLIOGRAPHY

Adams, James L. *The Care & Feeding of Ideas: A Guide to Encouraging Creativity*. Reading, Massachusetts: Addison-Wesley Publishing Company, Inc., 1986.

________. *Conceptual Blockbusting*. 3d ed. Reading, Massachusetts: Addison-Wesley Publishing Company, Inc., 1986.

Amabile, Teresa M. and Gryskiewicz, Stanley S. *Creativity in the R&D Laboratory*. Greensboro, North Carolina: Center for Creative Leadership, Technical Report Number 30. May, 1987.

Austin, James H. *Chase, Chance, & Creativity*. New York: Columbia University Press, 1978.

Blake, Robert R. and Mouton, Jane Srygley. *Productivity: the Human Side*. New York: AMACOM, 1981.

Clark, Charles H. *Idea Management: How to Motivate Creativity and Innovation*. New York: AMACOM, American Management Associations management briefing, 1980.

Harman, Willis W. *Creativity and Intuition in Business*. Menlo Park, California: SRI International, Business Intelligence Program Report Number 715, 1985.

Marsteller, William A. *Creative Management*. Chicago: Crain Books, 1981.

Maslow, Abraham H. *Motivation and Personality*. 2d ed. New York: Harper & Row, 1970.

Ray, Michael and Collins, James. "Creativity in Marketing"; "Organizational Vision." Text supplement to Marketing Issues briefing at Hewlett-Packard, Computer Systems Division, March 22, 1990.

Ray, Michael and Myers, Rochelle. *Creativity in Business*. New York: Doubleday, 1986.

Rosner, Stanley and Abt, Lawrence E., ed. *The Creative Experience*. New York: Grossman Publishers, 1970.

Susskind, Charles. *Understanding Technology*. Baltimore and London: The Johns Hopkins University Press, 1973

Von Oech, Roger. *A Whack on the Side of the Head: How to Unlock Your Mind for Innovation*. New York: Warner Books, 1983.

Network Management Today: Fact or Fiction?

Carole S. Crall
Hewlett-Packard
Fort Collins, Colorado

Introduction

Network management is becoming an increasingly "hot" topic whenever information systems are discussed. This is due, in part to the large percentage of a business' budget which is spent on computers, networks and personnel to manage them. Maximizing the efficiency of these systems and the people who use them is what systems and network management is all about.

This paper provides an overview of today's business problems which make effective network management solutions a top priority as well as introductory information on network management standards and solutions. Finally, future directions in network and system management are covered.

Defining the Problem

According to a study done by Infonetics, Inc. on the costs associated with network downtime, the average network goes totally or partially down over twenty-three times a year and is down for about four hours each time. Assessing the costs to an organization in lost productivity, revenue and direct expenses to get the network operational again is a difficult task. However, the study reported that on average, companies lose $3.48 million per year in lost productivity and $606,000 in lost revenue. At the same time, only $60,000 in direct expenses was expended. Lost revenue is usually associated with lost orders or inventory. Direct expenses is defined as personnel, software and hardware used during troubleshooting.

Taking all types of costs into account, businesses estimated anywhere from $10,000 to $200,000 the cost per hour of a network disability to their company. Using the results of actual cost studies and their own data, Infonetics has determined the average cost per hour of network downtime to be $30,000.

To compound an already out-of-control problem, Local Area Networks, LANs, are growing at a rate of 48% per year. Additionally, most of the expertise necessary to keep a network running is encapsulated in "home grown" tools and the knowledge of a few troubleshooters. Clearly a better way must be found or the number of people managing the network will also need to grow at a 48% rate. Good troubleshooters are scarce and currently command an average salary of $35,000 per year.

Because network problems dominate the typical day of network administrators, little bandwidth is left for critical activities such as planning for future growth. Companies are spending about $650,000 a year in upgrading their network and computer equipment to run faster and do more. Without proper planning, it's difficult to determine if this money has been well spent.

With the increase in distributed processing and proliferation of personal computers and workstations, not only is the number of devices attached to the network growing at a rapid rate, but the complexity and diversity of devices makes network operation and planning a challenge for every company. Figure 1 shows the type of network elements which may be present in today's network. From the least sophisticated components, the transmission links, to the very sophisticated networked applications, all network elements need to be managed effectively to maximize the investment in information systems. If all network elements were from a single vendor, had the same user interface and ran the same software, the problem of network management would never have become a problem.

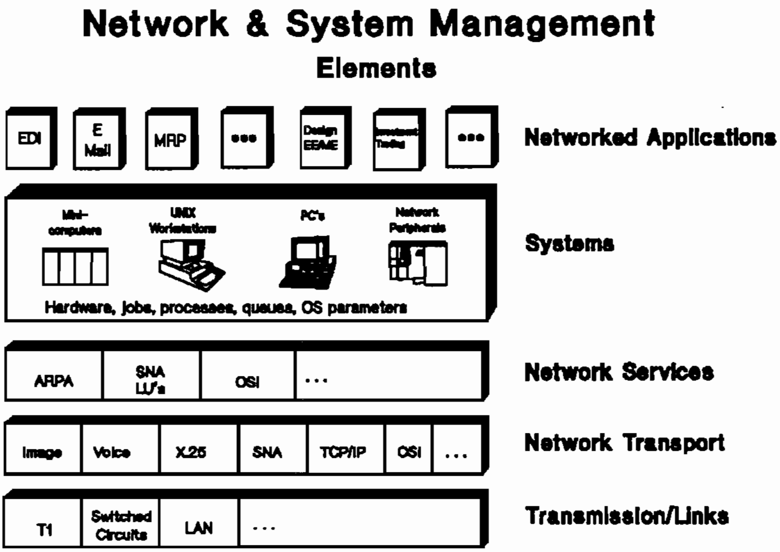


Figure 1

The State of Network Management Today

Network management can be characterized as using information technology to manage information technology. It is the remote monitoring and control of network elements from the transmission links to distributed systems and applications. And today, it's an afterthought.

The typical MIS department has a variety of tools currently being used to do network and system management. Many of these tools are instruments which require detailed knowledge to apply. Most of the tools have their own console and user interface. Although people are the most expensive and scarce resource in the network and systems management chain, companies depend heavily on the expertise they've developed to manage the network.

In order to get back in control of their networks and realize the maximum benefit from owning information systems, companies want and need to spend more money on tools from their hardware and software vendors and less money on highly-specialized, expensive personnel. Many of the tools available today are vendor-specific and come with a high learning curve so they only contribute to the problem of "home grown" expertise. Integrated, standardized tools which can be used quickly and easily by a variety of personnel are needed to allow businesses to effectively manage their networks.

In the last few years, the computer industry has been repeatedly hearing from customers that standardized, integrated network management tools are too late in coming. De facto network management standards have emerged and formal work has been done to define new network management standards to meet these needs. Vendors have responded with network management tools and platforms which conform to these new standards.

In order to discuss the different standards which have evolved and their uses, we need to understand the terminology and acronyms used to describe network management. First is the concept of **manager** and **agent**. Figure 2 depicts the International Standards Organization's (ISO) Open System Interconnection (OSI) network management model which represents the basic way network management is being implemented today.

The OSI Network Management Model

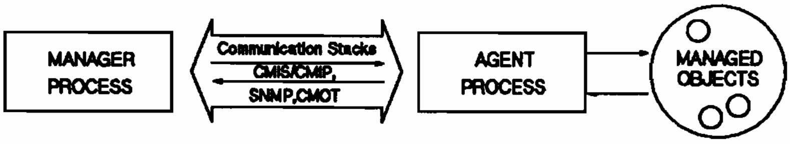


Figure 2

The model shows two processes at work, a **manager** and **agent**. These processes communicate via a standard protocol. We'll discuss these different protocols in more detail later. **Agent** processes collect information from and perform work on **managed objects.** A **managed object** is any component on the network which can be monitored and/or controlled. For example, modems, disk space, Ethernet cables, databases and repeaters can all be managed objects and are analogous to network elements we discussed earlier. Agents typically monitor and control managed objects which are local to them. **Managers** collect information from agents and request that agents perform work on their behalf. Managers present a centralized location for managing the network usually through a graphical user interface.

Communication between manager and agent and between agents and managed objects takes two forms: **operations** and **notifications**. Controlling network objects means performing **operations** on those objects which may include resetting counters, reconfiguring parameters or restarting the object, for example. **Notifications** are frequently called **events** or **alarms** and tell the manager or agent that some interesting condition has occurred. For example, if a system is about to run out of disk space, an **event** may be generated to tell the agent process of the condition. The agent would then report this condition to the manager process so the administrator could correct the situation from their centralized console.

SNMP, CMOT and CMIP

Communication between manager and agent processes is handled using formal and de facto standard protocols. In the last several years, three protocols have been developed and are endorsed by a variety of vendors and organizations. These protocols are discussed below. Because the Management Information Base, or **MIB** is a concept common to all three protocols we should first understand the role of the MIB in network management.

In order for managers and agents to communicate concerning managed objects and agents to communicate with managed objects, there must be an understanding of how these objects are specified. The **MIB** is the repository of these specifications. Included in the MIB are definitions of different network objects. These definitions include the objects' **attributes** including its name, **operations** the object understands and **events** the object can emit.

Today, the protocols described below use different MIBs to describe their networks (i.e. TCP/IP, OSI and a combination of the two). Work continues to converge the MIBs into one definition of all network objects. Successful completion of this effort is crucial if standardized, multivendor network management of a variety of network technologies is ever to be achieved.

CMIP

Because most of the formal standards work has involved defining the Common Management Information Procotol, CMIP, we'll begin our discussion with that protocol.

CMIP and CMIS, Common Management Information Service, reside over currently defined OSI transport protocols at the application layer of the OSI seven-layer model. The protocol and associated service provide a robust set of management capabilities including the ability to **get** and **set** an object's attributes, action commands, event notification, and the creation and deletion of MIB entries. These capabilities provide the foundation for implementing solutions in the five OSI-defined management areas: fault, configuration, security, accounting, performance.

Specific functions defined within CMIP/CMIS include:

- error reporting and information retrieval using defined error types, causes and severities.
- management service control including mechanisms for filtering events and requests.
- confidence and diagnostic capabilities such as test modeling and test control.
- log control to direct the logging of events.
- security management in the form of audit trails, alarms and security objects.
- accounting management to retrieve and control retrieval of accounting information from network objects.
- performance management through workload monitoring and threshold reporting.

CMIP has strong proponents and opponents in the industry. Because it is object-oriented it is very general and extensible. It also provides facilities to support information exchange between management systems. However, CMIP is still evolving and is untested for the most part. It is very complex and multi-layered and thus requires a great deal of information transfer to do its job. Finally, there are few OSI implementations today compared with TCP/IP implementations to manage.

As with any standard, implementations do not always interoperate. In 1988, the OSI Network Management Forum was founded to expedite network management interoperability through intervendor cooperation. The Forum now has over forty members and has done much work on implementation agreements for the CMIP/CMIS-based solutions. This will help ensure that as CMIP solutions are adopted, they will work together to solve the multivendor network management problem.

SNMP

The most widely endorsed and implemented protocol today is the Simple Network Management Protocol, SNMP. SNMP originally developed as a de facto standard used for managing TCP/IP (Transmission Control Protocol/Internet Protocol) networks. SNMP was recently promoted to full standard status by the Internet Activities Board which oversees the TCP/IP network community.

Because of its simplicity, SNMP agents can easily be implemented on network devices such as routers and bridges. Most of the burden of network management falls to the manager rather than the agent. Many implementations of SNMP agents are commercially available and there is SNMP agent software in the public domain. Manager applications written to use SNMP are also available commercially and in the public domain.

Three operations are defined in the SNMP specification: variable set, variable retrieve and event report. Unlike CMIP/CMIS, SNMP is not object-oriented and does not provide facilities for authentication. As much of the work is done from the manager via polling, it isn't clear whether SNMP is scalable or not. However, because of its simplicity it has been rapidly adopted by all major computer and device vendors. The next release of the Berkeley Software Distribution of UNIX* will include a version of SNMP for OSI network objects. This further muddies the water as to which network management solution to adopt for OSI network management.

Today the MIBs used by CMIP and SNMP are being developed and managed separately although the SNMP MIB is registered with ISO and efforts are in place to keep the MIBs in sync. The first version of the SNMP MIB contains definitions for 116 objects related to the TCP, IP, UDP (User Datagram Protocol), and EGP (Extended Gateway Protocol).

CMOT

The third protocol in our trio is CMOT or CMIP/CMIS Over TCP/IP. This specification was designed to smooth the transition from managing TCP/IP networks to managing OSI networks. It was envisioned that vendors would implement CMOT instead of SNMP. However, the momentum SNMP has in the TCP/IP community has proven difficult to overcome. Few implementations of CMOT are currently available although many vendors have indicated they will provide CMOT if their customers want it.

The Internet Activities Board approved a proposal early last year to split the SNMP and CMOT MIB work and that decision has further impacted the acceptance of CMOT. Added to the seeming lack of support by the TCP/IP community for CMOT is the push by SNMP supporters to squash it completely by implementing SNMP over OSI protocols as the migration path to management of OSI networks.

Technically, CMOT is the CMIP/CMIS specification implemented over OSI layers five and six and then over the TCP/IP stack. A "thin" presentation layer provides the translations necessary between OSI and TCP/UDP.

Commercial Network Management Solutions

Major computer and system vendors offer network management solutions which are designed to solve a wide variety of problems. This section presents a brief introduction to several vendors' solutions and is intended only to provide a foundation to be used to pursue further information if desired.

Solutions are commonly placed into two different groups by industry analysts: enterprise management solutions and LAN/Internet management solutions. Enterprise management refers to the integrated management of corporate backbone networks including both the voice and data components. LAN/Internet management is focused on the data networks within the overall corporate backbone and therefore, does not handle any of the voice network management.

NetView from IBM is one of the oldest network management solutions in place today. NetView is both IBM's network management strategy and product family. NetView includes products which run on systems from mainframes to PCs. IBM endorses SNMP and is a member of the OSI Network Management Forum.

---

* UNIX is a registered trademark of AT&T in the U.S. and other countries.

Digital Equipment Corporation 's network management strategy is defined in their Enterprise Management Architecture, EMA. Digital currently has several system and network management products for their product lines. Digital endorses SNMP, CMOT and is a member of the OSI Network Management Forum.

Unified Network Management Architecture, UNMA, from AT&T is their network management strategy. Today, AT&T's main product is the Accumaster Integrator which is based on OSI network management standards. AT&T was a co-founder of the OSI Network Management Forum and endorses SNMP.

HP OpenView is Hewlett-Packard's network and system management strategy as well as product family. Several HP OpenView products exist today to solve a variety of management problems. HP was a co-founder of the OSI Network Management Forum and endorses SNMP and CMOT.

SunNet Manager from Sun Microsystems is an SNMP-based product within the SunNet family of networking products. SunNet Manager is focused on management of TCP/IP environments and is a LAN/Internet management solution.

Open Management Architecture (OMA) from 3COM incorporates support for multiple protocols: SNMP, CMOT, and CMIP. OMA is also a LAN/Internet management solution.

Future Directions in Network Management

As you can see, the network management problem is complex and while there are a variety of solutions available in the marketplace to solve network management problems, none will solve all the problems. Because of the diversity of networks and their uses, customers will need to customize management solutions and often write their own solutions.

Industry analysts have spent a great deal of time evaluating different vendors network management offerings. They predict several trends in the coming years. First, SNMP and CMIP are being and will continue to be widely adopted in lieu of vendor-specific network management protocols. While there is a proliferation of consoles and management solutions today which are unintegrated, future solutions will still have multiple consoles, but all products will work together to solve the overall network management problem. Vendors and system integrators will continue to embrace well-defined network management platforms as the basis of their solutions.

The 1980's can be characterized as a time of fragmentation in the network management world, where vendor-specific solutions dominated the marketplace. In the early 1990's, integrated network management will become a reality through adoption of standards-based platforms and work by such organizations as the OSI Network Management Forum. The next step beyond integrated network management is automated network management whereby common management tasks are performed by software rather than people. By the late 1990's, knowledge-based management will be a reality and the problem-solving capabilities of our expert administrators today will have migrated into software, greatly reducing staffing requirements.

References

1). **The Cost of LAN Downtime**, Infonetics, Inc., 1989.

2). **Multi-Vendor Network Management**, James Herman of Northeast Consulting Resources.

INFORMATION MANAGEMENT TECHNOLOGIES INTO THE 1990'S

Orland J. Larson
Hewlett-Packard Company
19091 Pruneridge Avenue
Cupertino, California 95014

ABSTRACT

Enterprises have come to depend more and more on the accessibility, accuracy and timeliness of information. During the next few years, new database technologies will provide solutions to many of the problems and difficulties facing today's MIS and user departments.

This paper begins by reviewing the changing world of information management and the challenges facing MIS. This will be followed by the major trends associated with information management over the next several years. These include the continued acceptance and importance of relational database technology, the increased interest in distributed database applications and the emergence of the new object-oriented database management capabilities. Finally, this paper will address the significance of these technologies in the cooperative computing environment.

INTRODUCTION

THE CHANGING ROLE OF DATA PROCESSING

Before looking forward into the 1990's, we should first look back at how data processing has evolved over the past twenty years.

The general trend of the 1970's was the use of centralized computers and resulted in systems that were often difficult to use, inflexible and usually did not meet the end user's needs. Database management systems (mostly hierarchical and network) became widely used and provided the basis for on-line, interactive applications. In addition, the computers and operating systems provided programmers the capability of developing applications on-line, while sitting at a terminal and interactively developing, compiling and testing these applications. The end users were also provided easy-to-use, on-line inquiry facilities to allow them to access and report on data residing in their databases.

During the 1980's, the emphasis was on the decentralization of data processing. This includes the proliferation of personal computers which has resulted in both the "islands of automation" and the corresponding "islands of information" problems. This in turn resulted in reduced control of corporate data for the MIS department. In addition, relational databases became commercially viable and experienced wide acceptance even though performance was often an issue. Relational database performance has now improved significantly; and they are currently proving effective in on-line transaction processing (OLTP) environments. Software tools such as 4th generation languages (4GL's) continue to be used successfully as an effective way of developing applications through the concept of information systems prototyping. This required that the end user be more involved in the development of systems and has resulted in more effective systems that meet the users' needs. This has helped to reduced the backlog of applications but usually also has contributed to the "islands of automation" problem.

As we move into the 1990's, relational database will continue to gain wider acceptance. It is the enabling technology and the basis for distributed database management, which provides transparent access to data which is distributed over several sites.

There are also new technologies called object oriented database (OODB) and object oriented programming systems (OOPS), which will manage more complex data structures and will result in improved programmer productivity and more flexible systems.

An additional technology, cooperative processing, is evolving which will help integrate those "islands of automation" back together and allow for the data and programs to be accessed and shared in a cooperative computing environment.

THE IMPORTANCE OF SQL

According to a recent Gartner Group Report, in 1988 only about 7% of the applications developed used relational database or Structured Query Language (SQL). However by 1992 their prediction is that 65% of applications developed will use SQL.

There is no doubt that SQL will be the basis for applications developed in the 1990's. One of the main advantages of SQL is data independence or the immunity of applications to changes in storage structure and access strategy. Another main advantage of SQL is the simplicity of the underlying relational model which is the easiest to understand - at least at the most basic level. In this model, data are represented as tables, with each horizontal row representing a logical record and each vertical column representing one of the attributes, or fields, of the record.

The following are the key points associated with relational technology:

* Relational concepts are easy to understand and use.
* SQL is a multifunctional language
  - Database definition and creation
  - Data retrieval and manipulation
  - Authorization and security
  - Transaction management and recovery
  - Database environment management and restructuring
  - Interactive and programmatic use
* SQL allows you to specify which information you want - not how to retrieve it.
* SQL increases programmer productivity and raises programming closer level of problem solving.
* Data independence is ensured and minimizes maintenance of programs
* Data access can be automatically optimized as the DB structure changes.
* The DBA has unprecedented power and control over the database.
* New systems can be implemented much faster.
* SQL assists in cross-system connectivity.
* Relational databases provide a cost effective, powerful solution.
* Basis for a true distributed database environment.

There are, however, some areas of SQL that need improvement. The current SQL standard is missing many important features and many of the standard features are implementor defined. In other words, no vendor fully supports the complete "standard" and no two SQL implementations are exactly alike. These inconsistencies will lessen as new levels of standards evolve. Organizations like the SQL Access Group are also working hard to resolve these problems.

As we enter the 1990's, relational database is becoming the dominant technology in today's information management marketplace. There are several enhancements planned to improve functionality and performance. It eventually will be appropriate for most applications and gain wide acceptance by all users.

Relational databases can improve the quality, control and accessibility to your organization's extremely important and valuable information resources. It can result in an improved competitive position by aiding business analysis that can help to determine ways to improve products and services.

Unlike non-relational database environments, relational databases adapt easily to dynamic business requirements. In addition, unrestricted access to important data means better information for more effective decision making.

Relational database can also have a positive effect on many MIS development environments by reducing the application backlog and reducing the time and cost required to develop applications. The improved database flexibility and ease of change can also result in a significant reduction in the maintenance of applications.

Overall, the use of relational technology can increase the MIS professional's effectiveness and productivity, which results in improved user satisfaction and confidence. Choosing relational now will position your organization to take full advantage of the technological advances of the 1990'S.

DISTRIBUTED DATABASE

One of the hottest topics in the commercial database world is the growing trend towards the use of distributed database management systems. After many years of research, distributed databases are becoming more viable. However, there is still much to be done to provide more than just read access to distributed data. Chris Date, one of the world's leading experts on relational database, recently presented a paper (see reference 3) in which he provided a working definition of distributed database. "A distributed database system is a system involving multiple sites connected together in a communication network, in which each site is a database system in its own right, plus a user at any site can access any data in the network exactly as if the data were all stored at the user's own site. Thus, a DDB is a virtual DB whose components are physically stored in a number of distinct real databases at a number of distinct sites."

Chris Date follows this working definition with an "alternate" or more elaborate definition. "A distributed database system is a system that allows an arbitrary collection of relations, from an arbitrary collection of databases, on a variety of different machines, running a variety of different operating systems, connected by a variety of different communication networks to function as if they were all stored in a single database on a single machine. The user is completely insulated from all details of distribution."

Distributed databases can allow the structure of the database to mirror the structure of the company, while simultaneously solving the "islands of information" problem. Some additional advantages include local control of local data, accessibility to remote data, increased capacity, incremental growth, data availability, efficiency of storage, flexibility and cost effectiveness.

There are also some potential problems or disadvantages associated with distributed database such as the complexity of implementation - but this is the vendor's problem. Some additional potential problems include the problem of how to design systems for distributed environments, the complexity of administration and control, the impact on local operations, the political problems dealing with the ownership and protection of the data and the possibility of a node or line "crash". In addition, solutions that are appropriate in a centralized environment may frequently not be appropriate with distributed systems.

Chris Date is the author of 12 rules of a distributed database system. He begins with the fundamental principle or "rule zero" that states, "To the user, a distributed database system should look exactly like a nondistributed system". The subsidiary rules follow:

1. Local autonomy
2. No reliance on a central site
3. Continuous operation
4. Location independence
5. Fragmentation independence
6. Replication independence
7. Distributed query processing
8. Distributed transaction management
9. Hardware independence
10. Operating system independence
11. Network independence
12. DBMS independence

These rules are fairly self explanatory and will not be expanded in this paper. No current DBMS vendor adheres to all of these rules. A few vendors claim adherence to rules 1-8 and almost none to rules 9-12.

David Wilde, Project Manager at Hewlett-Packard's Database Lab, recently spoke at the San Francisco INTEREX Conference and presented HP's phased approach for developing a distributed database management system. The following is a summary of that approach:

* REMOTE DATABASE ACCESS (ALLBASE/NET) - Program can read/update a remote DB without coding for communication and remote processes.

* FOREIGN DATABASE ACCESS (e.g. IBM's DB2) - Program can access multiple vendors databases without coding for DBMS differences.

* MULTI-REMOTE DATABASE ACCESS - Program can read and update more than one remote database at a time.

* SNAPSHOTS - Enables a user to copy all or part of a table from one

database to another, optionally this table could be refreshed.

* PARALLEL QUERY EXECUTION - Ability to execute multiple queries to multible databases at the same time.

* DISTRIBUTED UPDATES WITH TRANSACTION MANAGEMENT - Enhanced transaction management to support updates to multiple DB environments.

* REPLICATED DATA - Enhanced availability and performance through multiple copies of data, with automatic synchronization of copies.

* PARTITIONING - Enhanced performance and availability through partitioned tables.

Some of the issues regarding distributed database that will have to be addressed in the 1990's include:

* Distributed query optimization and decomposition
* Fragmentation, recombination and optimizability of data
* 2-phase commit and recovery
* Referential integrity across sites
* Management of replicated/partitioned data
* Controlling authorized user access
* Update synchronization
* Degree of Transparency
* Flexibility to move data around the network
* Cost of mainframe vs. mini vs. micro MIPS
* Provision of foreign (non-HP) DBMS gateways (gateways are a way of processing data in a foreign DBMS or file system)

There are some factors that will help propel the distribution of data. These include company mergers, the downsizing of computers, the increased database needs and the general industry push toward distributed and cooperative processing. A company with distributed operations will gain competitive advantage through the support of distributed databases.

OBJECT-ORIENTED SYSTEMS

Each decade, one or two key advances emerge to change the practice of software development. Object-oriented systems and methods are rapidly entering the mainstream of software engineering and systems development. Leading consultants are heralding object-oriented approaches as one of the most important trends to affect businesses in the 1990's. But even among its strongest advocates, disputes abound over key issues, content, and definitions of the object-oriented approach. Object-oriented technologies are moving out of the academic world and into the business world.

With the object-oriented approach, processes revolve around the data, not the other way around. Using the traditional approach, programs are structured around data rather than procedures. For example, when using a traditional programming language, parameters are used to pass data structures and values between routines. The object-oriented approach attaches routines to data structures. In other words, the behavior of the data is kept with the data. This is called encapsulation.

An object-oriented programming language allows the programmer to define and manipulate objects. Some object-oriented programming languages are extensions of classical languages - C++ and ObjectiveC are in this category. Others are brand new languages, eg. Smalltalk and Eiffel.

An object-oriented DBMS also supports the definition and manipulation of objects, plus providing the classic DBMS functions of persistent storage, transaction management, concurrency control, security, backup and recovery.

A "message" is an important concept used with the object-oriented approach. It is defined as a request sent to an object to change its state, or to return a result to the sender. Objects respond only to well-defined messages. The only information needed to use an object is knowledge of the messages it can receive. An object-oriented program is a flow of messages among cooperating objects.

Messages ensure the modularity of a system. To interact with any object, you only need to know what messages to respond to, not how the object is represented. Knowledge of how an operation is accomplished is of interest only to the programmer responsible for the definition of the object itself. Messages make an object's functionality available to other objects, while hiding the implementation details.

Maintaining and modifying software has been a real drain on programming resources. Maintenance programmers must understand a complex system well enough to fix its problems or enhance it. However, these programmers often did not develop the code, and often operate without adequate documentation or guidance. Changes often introduce new, unanticipated problems to the system. A programmer working with existing code must read and understand it; this may require a mental translation back into the original design specifications, which is extremely difficult for complex or poorly-coded systems. An additional concern is that programming languages allow unchecked access to data structure internals.

Objects can dramatically improve the problem of maintenance. Modularity and encapsulation limit interdependence, allowing changes that do not disrupt the rest of the system. Objects' natural organization make it easier to learn and understand relationships between parts of a system. The original programs are easier to write and debug and fewer errors occur. Programs read like designs, making changes clear and easy to make. Reuseability is an important advantage of using objects and libraries of these software components provide leverage. What has been written once need not be written again. Model features like inheritance allow existing components to be incrementally modified to suit changing needs. Together the representational advantage of encapsulation and the features of inheritance dramatically improve software development and can greatly improve programmer productivity.

Libraries of high-quality, tested software components will radically alter the way software is written. Software will routinely consist of a series of software components glued together. Application programming will no longer mean rushing to a text editor to begin coding. It will require understanding the capabilities and restrictions of available components, plus knowing how to combine them.

Some of the challenges facing object systems include the time it takes to learn about existing libraries of software components. Programmers also may resist accepting this new approach. Objects also consume more resources, however emerging 80486 PC's and RISC workstations will help to alleviate this problem. Applications with promising potential include: prototyping, user interfaces, graphics, telecommunications, computer aided design (CAD) and computer aided manufacturing (CAM).

Object oriented products are still in their infancy and the commercialization of object-oriented technology has barely begun. Now more suitable for advanced technology projects, object database systems should become viable for commercial projects over the next few years and widespread adoption by the mid-1990's.

Hewlett-Packard has developed a prototype of an object-oriented database management system (OODBMS) called Iris. Development started in 1985 and Iris has been presented and demonstrated at several major conferences in the past few years including: SIGMOD (Special Interest Group on the Management of Data) in June 1988 in Chicago, OOPSLA (Object-oriented Programming, Systems, Languages and Applications) in September 1988 in San Diego, DB/EXPO in February 1989 in San Francisco and at the Patricia Seybold Forum on Object-oriented Technology in April 1989 in Boston.

Object-oriented environments herald the dawn of new programming paradigms. Business people will be empowered to perform tasks that, in the past, required professional programmers. Programmers will be empowered to design complex applications in smaller, modular, more fool-proof pieces.

Neither end users nor application programmers will need to concern themselves with the mechanics of networking, peripheral support, or file handling. Object based architectures lend themselves to the creation of a much richer information environment. Digitized voice, music, images, video clips, and animation will begin to populate our information systems.

COOPERATIVE COMPUTING

The environment in which today's business must operate is changing quickly and becoming more complex. To meet the challenges produced by this changing environment, organizations need greater amounts of information to make the key decisions required for success. Keeping pace with the rapid change that is occurring means gathering information and making decisions faster than ever before.

As we look at the computer industry, we are about to embark upon the next revolution in computing. This is not a unique case. There have been multiple revolutions in the computing industry and computation in general, going back to the early days of mechanization, tabulation and so on, through the first computer mainframes. The mainframe was a very centralized processor, still oriented toward batch and was really a carry-over from punchcard tabulation systems. The next move was into mini-computers, and distributed processing followed by the personal computer revolution. This caused an explosion of workstations and personal computers on people's desks, which fueled distributed processing and distributed computing.

Then came the communications revolution, where the objective was to integrate all this computing power in the corporation in a way that moves information around and enables different types of devices to participate in the solution of business problems. This has become the basis or launching point for the next revolution of cooperative computing.

Cooperative computing is the notion of tying together all the information computation resources in a corporation into a single entity, and making all those things interact in some efficient manner, transparent to the end-user in such a way that each user has access to all the information computation resources in the network as though they were local to the user's workstation.

NewWave Computing is Hewlett-Packard's implementation of the industry trend towards cooperative computing. It includes HP's vision of the future of computing: a network of heterogeneous computers that can work together to solve a single problem and are extremely easy to use.

To support cooperative computing requires a strategy that addresses the technological implementation needs. The NewWave Computing Architecture brings together systems and servers, easy to use workstations, industry standard networking and perhaps most importantly, the integration of the three through innovative software.

This paper has addressed many of the important technologies that are essential to the cooperative computing environment. ALLBASE/SQL is HP's strategic DBMS of the future for cooperative computing. ALLBASE/SQL runs under both MPE and HP-UX. The application development environment for both of these platforms is a powerful 4th generation language called ALLBASE/4GL and an easy to use end-user oriented inquiry and reporting facility called ALLBASE/QUERY. ALLBASE/SQL will also be the basis for distributed database technology in the future. ALLBASE/NET is the first phase of HP's distributed database technology providing remote data access and uses HP's networking capabilities.

ALLBASE/DB2 CONNECT allows ALLBASE/SQL to access (read and write) DB2 databases on IBM mainframes. Access to DB2 from both PCs and HP3000s will be supported from ISQL (Interactive SQL), Info Access (PC-based product) and ALLBASE/QUERY. ALLBASE/DB2 CONNECT is the flagship product by which we are launching connectivity to non-HP database servers, and extending the PC-HP3000 environment to include IBM mainframes. ALLBASE/DB2 CONNECT will be available in late 1990.

The concept of "objects" in the HP NewWave environment is similiar to the object-oriented capabilities described in this paper. The Object Management Facility (OMF) is one of the main components of the HP NewWave environment. The OMF tracks all data in the PC, whether it be text, graphics, spreadsheets, scanned images, even voice. These objects are represented as icons which can be combined into compound documents containing different types of data.

The OMF allows users to create "hot-links". Hot-links allow users to share data between different reports, memos, even file folders. When data is changed in one place it is updated automatically in other shared documents throughout the system.

The OMF binds applications and data together to form "objects". By double-clicking on an object, you can simultaneously load the application and call up the desired file, eliminating the need to find file names and their directories.

SUMMARY

The important technologies briefly addressed in this paper: SQL, distributed database management systems and object-oriented systems are extremely important to the future of the cooperative computing environment. HP's NewWave Computing strategy is focused upon helping our customers meet the challenges of today's changing business environment. We believe that these technologies and strategy, coupled with HP's commitment to industry standards and reputation for high quality, reliable systems, can provide our customers with the solutions they will need into the next generations.

REFERENCES

Connors, T. and Lyngbaek, P.,"Providing Uniform Access to Heterogeneous Information Bases",Software Technology Laboratory,Report STL-87-16, Hewlett-Packard Laboratories,December 4, 1987.

Date, C.J.,"Distributed Database Technology", Proceedings DB/EXPO'89, San Francisco,CA,January 31-February 2,1989.

Eunice, J.,"Leaders in Object Databases",D.H. Brown Associates, Inc., December 1988,400 Benedict Avenue,Tarrytown,NY,10591,(914)631-7859.

Fishman,D.H.,Beech,D.,Cate,H.P.,Chow,E.C.,Connors,T.,Davis,J.W., Derrett,N.,Hoch,C.J.,Kent,W.,Lyngbaek,P.,Mahbod,B.,Neimat,M.A., Ryan,T.A.,Shan,M.C.,"Iris: An Object-Oriented Database Management System",ACM Transactions on Office Information Systems,5(1),Jan 87.

Larson, O.J.,"Strategic Importance of Relational Database Technology", Proceedings INTEREX Computing Management Symposium'89,Nashville,TN, March 8-10, 1989.

Perreault,R.,Rotzell,K.,"Object Database Management Systems",INTERACT, April 1990,pp.114-126.

Schussel,G,"Database Developments in the 5th Generation",Proceedings 1989 Database and Cooperative Processing Symposium,Washington,DC, March 20-22, 1989.

Stein,J., and Maier,D.,"Concepts in Object-oriented Data Management", Database Programming and Design,1(4), April 1988, pp.58-67.

Tash, J.,"Application Architecture and the Strategic Role of Relational DBMS",Database Decisions, Inc.,41 Marcellus Drive, Newton,MA,02159, (617)332-3101.

White, C.,"The Importance of SQL",Proceedings DB/EXPO'89,San Francisco CA,January 31-February 2, 1989.

Wilde,D.,"Distributed Data Management From HP",Proceedings INTEREX HP Users Conference,San Francisco, CA,September 11-14, 1989.

Stress Management made 'relatively' Easy

Robert M. Gignac
C&M Products Limited
189 Bullock Drive
Markham, Ontario (Canada)
L3P 1W4
(416) 294-9570

Abstract:

Stress management was the rage of the 80's, and will continue on into the 90's. Why? I propose it is for two reasons: 1) The environments we work in haven't changed, and 2) We haven't been listening well enough. Stress management is much like dieting, we all know we need it, and that it is good for us, but we lack the motivation required to break the inertia preventing us from doing something about it.

This paper will explore what can be done about stress management without having to spend: 1) A great deal of time, or 2) A great deal of money. As a matter of fact, some of these techniques can even be used at your desk, which should make applying them to the 'real' world even easier.

Introduction:

When I started to put this paper together, everyone I talked to wanted to know about the 'relatively' comment in the title. Why wasn't this paper called "Stress Management made Easy", or "Easy Stress Management", or something along those lines. Why 'relatively'? The answer is simple. Because it isn't 'easy'. Very few things worthwhile are. The problem that exists is with perception (as we shall see later...). If you tell someone it is difficult, they enter into it expecting to fail. If you tell them it is easy, they expect to put forth no effort to achieve benefit. In reality, when it comes to stress management, neither of these situations is the case, hence 'relatively' in the title. This out of the way, let us proceed to find out just how 'relatively' easy it is.....

What is Stress?

Stress has many definitions, including the following: "strain, mental or physical tension", "actions or situations placing special physical or psychological demands on the person", "the response of the body to any given demand". It is believed that stress often results from environmental situations that require behavioral adjustments - situations ranging from petty daily annoyances (traffic jams, late buses, broken fingernails) to events such as significant illness, death of a spouse or divorce.

Job dissatisfaction, crowding and commuting may all contribute to stress. On-the-job stress may have any number of causes. Your job may require a great deal of effort and rapid decision making; its requirements may be complex or ambiguous; or there may be the threat of unemployment, pressure for advancement, job role conflict or pressure to conform (ie: company policy, dress codes, etc.). Unsatisfactory work conditions, including repetitive boring work, inconsiderate supervisors, rotating shifts, etc, call for coping responses from each individual. Daily hassles such as these are regular and often permanent, and can obscure the severeness of the cause.

It is easy to see how stress plays a role in our working and personal lives as stress is created by the adaptation of our body and mind to change. In today's workplace, with its uncertain economic conditions and continuing technological advancement it often seems that the only constant is change.

What is Stress Management?

Stress management is the process of taking those elements in your life that create stress for you and learning to: 1) adapt to them, 2) reduce the severity of stress these elements cause you, or 3) learn how to avoid the stressors if the first two cannot be accomplished. Stress management will not eliminate stress from your life. First, it would be next to impossible to perform such a task. As desirable as it may seem, removing such stressors as traffic jams, air and noise pollution, overcrowding, work deadlines and inflexible company policies,. is just not possible. Second, even if we could accomplish it, completely eliminating stress is not desirable as some stress is a requirement for daily living. Stress provides us with the maturation required to accomplish tasks, to change things and stress can extend our abilities to the point that we often do our best work when under non-debilitating pressure. Many examples of this can be seen in the sports world where records are rarely set in routine practise sessions, but often under intense competition (ie: Olympics or World Championships).

What steps should we take to help us manage stress? This paper will focus on two: Content and Concept.

Stress Management - The Content

The content of stress management is the quantifiable portion, filled with specific tangible data. Exercise, a healthy diet, rest/relaxation and developing appraisal skills are the keys. Not only is the content good for us, it works. If we were to exercise regularly, eat a healthy diet, take proper vacations and 'time out' at appropriate intervals, in addition to having regular medical checkups, we would all be much better off. If its so easy, why don't we do it?

Time, lack of motivation, fear of the unknown perhaps? The reasons we choose to avoid doing what is intrinsically good for us are as varied as our individual personalities. What we must do is breakdown the barriers that prevent us from helping ourselves.

Exercise

The mere mention of this word is enough to make many people break into a sweat - a cold one at that. Exercise conjures up images of hot steamy weight rooms, sweat, grey flannels, and pain. I'm here to tell you that it isn't so. Exercise doesn't have to be gruelling or become life's sole obsession in order to be beneficial. The Nike shoe company did everyone a big favour with their 1989 advertising campaign - "Just do it!". Walk, swim, bike, jog, skip rope, lift weights, play baseball, any sporting activity that brings you pleasure. If you've chosen an exercise that you find distasteful, drop it. Rather than do a little of a distasteful exercise, try more of an enjoyable one. In addition to this, the word 'moderation' comes to mind. Think of when the major fitness clubs do their heaviest promotions. Generally in January, just after we have feasted for the holidays only to find we shouldn't have. Suddenly, the clothes that fit us so well in October no longer feel the same. The realization then hits that we should do something about it. Now. We join the fitness club, buy the new sweats & shoes and are away to the races. Do we start with the low impact 15 minute aerobic class? No way. We jump right into the 45 minute, high intensity aerobic class because we are going to get ourselves back in shape in a hurry. What generally follows are days of slow agonizing movements and a sudden loss of our resolve with the comment "See, I told you that exercise is no good for me".

If I hear any complaint about exercise, it has to be the major complaint of life today - "I don't have enough time". When it comes to time, all things are relative. If you'd like an Arnold Schwartzenegger physique, 15 minutes a day won't do it. For the purpose of making us a little less sedentary, and increasing our energy levels, 15 minutes a day would be a great start. Can't get away from your desk, you say? That's okay, you can start while sitting at it. How? Easy. Keep a pair of sponge rubber balls or exercise handgrips in your desk drawer. On hold while on the phone? Use them. When taking that 5 minute thought break while working on a difficult problem - use them. It may sound easy, but try squeezing a sponge rubber ball for five consecutive minutes and see how your forearms feel, especially if you aren't used to it. Another useful office technique is to keep a pair of 'softweights' (for the ankle or wrist) in your desk. Wear the ankle weights as you walk around the office. The walking you are doing is useful as it is, the extra resistance caused by the weights will make the movements even more beneficial. (At this point I should add that the ankle weights idea is predominantly directed at the males, I've been told they tend to look a tad unsightly with a skirt and high heels....)

In short - "Just do it!" Do whatever exercise makes you feel good and causes your heart rate to increase beyond its sedentary level. Remember that every little bit helps.

Diet

Diet covers two separate topics - the act of dieting and the concept of eating healthy. The act of dieting has been covered enough in other publications that I won't cover it here. Good nutrition on the other hand, is an area within the scope of this paper. A healthy diet is a great step toward creating better health that will allow us to face stressful situations with fewer adverse effects. Given the fact that we already know this, it is hard to believe what we do to ourselves on a daily basis in the workplace. If we are going to do something nice for our fellow office workers, what do we bring in with us, a dozen pieces of fruit, or a dozen donuts? Of course we bring in the donuts. We all have coffee or soda machines at work, it is harder to get milk or fruit juice (though most soda machines today have some selection of fruit juice or diet soda). We could bring a healthy home-made lunch to work with us, but we want to be 'one of the boys/girls' and we dine out, generally on 'fast-food'. In order to make ourselves feel better while we are doing this, we choose what we feel is the best dietary alternative from the menu. We tend to order the chicken sandwich over the hamburger at Burger King (no slight intended to McDonald's or Wendy's fans) because it is better for us, right? Better tasting perhaps, but not from a calorie/fat per serving perspective.

Type of Sandwich	# of calories	tsp of fat
Hamburger	255	2.50
Fish	435	6.25
Chicken	690	10.50

I am by no means recommending we become vegetarians, and pursue life on a 1000 calorie/day diet. It is however, important that we give a little more thought to what we eat on a daily basis because good physical health will put us in a better position from which to deal with stress.

Relaxation

A popular piece of advice runs "Just relax. Keep cool. Take it easy". In reality, this is often easier said than done. Many people complain that they don't know how to relax. Often the word 'workaholic' describes them well - people so preoccupied with their work that even their vacations are stressful.

As difficult as it may be to believe, relaxation is a skill and as with any skill it requires practise. There are audio-tapes available that will allow you to practise relaxation techniques such as muscular contraction and deep breathing. As your skills develop, relaxation through slow deep breathing can be used as an active coping skill to control physical tension and stress on the job. Mental relaxation is also a key part of the process. Some of us relax by engaging in strenuous exercise (see section above), others by becoming absorbed in a task (gardening, knitting), still others by meditation, yoga or progressive muscle relaxation.

Learn to Appraise Situations

Appraisal or perception is an important element in determining how we view stressful situations. 'Cognitive Appraisal' refers to an individuals judgement of the demands and constraints in any given situation and the options at our disposal for dealing with them. Appraisal can be viewed as a two stage process - primary and secondary.

During primary appraisal we view the situation to be either relevant or irrelevant, challenging or threatening, of possible benefit or harm. These decisions are often made without our thinking about it - at an almost unconscious level. The primary appraisal answers the question "Am I ok or am I in trouble". How did the person on the other side of my desk approach with this problem? Was it "Hey Bob, do you have a minute? I think there may be a problem with the XYZ program". Or was it "BOB, WE

HAVE A SERIOUS PROBLEM HERE, THE MACHINE'S ON FIRE!". Depending on how we view an event, as a threat, potential loss or a challenge, will determine the degree of our stress reaction. Personal beliefs, goals, values and commitments will all factor into our primary appraisal of events.

During secondary appraisal we make judgements about the adequacy of our available coping skills, as they pertain to the situation. It is here that we answer the question "What do I do about this situation". We evaluate our coping strategies in terms of their cost and the probability of their success. This decision is based on our past performance, self-confidence, and material resources available. Secondary appraisal, like primary appraisal, occurs automatically. Our attempts to cope with a situation cause it to change, requiring further reappraisal in an endless cycle until the situation is resolved.

Areas that we can work on in the appraisal process center around our ability to see situations for what they really are. Perception is indeed reality. When we view a situation, we act upon it based on how it appears to us, even though we may not be seeing the situation in the proper light. An analogy here is to ask children in a family of 5 how life was growing up and to describe their relationships with their siblings. There is a very good chance that you will end up with five entirely different stories, but each individual will view theirs as the truth. Taking the time to realistically appraise situations can ensure that we make better decisions when acting upon this information. If possible, take yourself out of the situation and try to view it as an outsider would, or perhaps as the other person in the situation does. Developing empathy with the other people you deal with on a regular basis can help to ensure you view situations in the best possible light, and ensure that your reactions will be appropriate.

Stress Management - The Concept

The concept of stress management is the second half of a cohesive model to deal with stress. The content discussed earlier are things that we need to do to make us feel better, but if we abhor exercise and despise fruit and vegetables (even President Bush won't eat broccoli) all the extoling in the world on my part isn't likely to get you to change. Also keep in mind that the content alone will not solve your stress problems. Therefore the concept portion will focus on the personal side of stress management- the concept of power, dealing with difficult people, viewing energy as pie, values theory, salami theory, dropping problems and maintaining balance in the face of stress.

The Concept of Power

One of the main criteria when dealing with stress is power. Powerlessness in a situation increases our stress because we feel that actions and consequences are out of our control. Power is not a function of position, regardless of what your company's organization chart may indicate. People obtain/lose power as a function of their relationships with others. It is in this way that children have power over parents, when it should be the reverse. Dependency gives people power because once someone becomes dependant, the person feeding the dependency must continue to do so, or suffer the wrath of those dependant on them. Powerlessness is regularly described as having a lack of available choices. Ask someone who puts up with a poor work situation or an overbearing supervisor and they respond with - "I don't/didn't have any choice". The truth, however ugly, is that we always have choices- many of them- but invariably we don't like them. When we don't like the choices available, we just don't seem to have any choice. In order to regain power is stressful situations, we must make the difficult choices, even ones that are distasteful because they are often the key to removing ourselves from a stressful situation.

Dealing with Difficult People

Invariably, our workplaces tend to force us into personal contact with others on a daily basis. The majority of this contact will be interesting, enlightening and make work enjoyable. There will be others however, that will make work less than pleasant, perhaps downright unbearable. Let's call these 'difficult' people. Rather than setting our Star Trek phasers on 'stun', how can we deal with difficult people? First, we have to realize that is is the difficult people that cause us to be powerless in work situations. Not because they take our power away, but because we give it to them. Second, realize that you can't enlighten unconscious people. While not unconscious in the physical sense, they are harder to deal with because we tend to see these 'unconscious' people as egocentric, controlling types who 'just don't get it'. Are they conscious? Of course. Do we see it? No. To make matter worse, the more burdened we become, the worse these difficult people seem to be. Under stress our perceptions change, and however incorrect, our perceptions are indeed our reality in any given situation.

Take as an example the habitually late employee. Our management workshops teach us to document this lateness and invariably we confront them with this documentation. Faced with this documentation, they will generally chose to ignore it, or give you a 'solid' reason for every day that they were late. They will try to turn it against you - "Don't you have anything better to do with your day than keep track of my time?". Or perhaps - "well, yes, I've been late a few times, but you were late last Thursday". The key to remember when dealing with difficult people is that you can't get people to solve problems that aren't theirs and you aren't helping them by taking on their problem. If the tardy employee doesn't view it as his problem, he will never take responsibility for it. As we will find out later, ownership of a problem is the key to starting to solve it.

Pie Theory

Pie theory is an attempt to describe how we gain, lose and use energy on a day to day basis. Envision your energy total as a pie - you are allowed to cut/slice your pie any way you like, but you only get one pie per day. After all the day to day details of work, home and play, you'll find you rarely have much, if any, left for yourself. The key, therefore, is to spend our pie wisely. What is the best way to do this? Spend you energy on conscious people, on tasks and goals, and reserve some for yourself. If you are an 'average' individual you probably tend to give most of our energy to those people who create problems for us. Think of how our school systems work. Where do schools spend the majority of their energy, on 'F' students or 'B' students? By focusing on the problems at work and on difficult people, we end up with less energy to give to those who are most important to us.

Dealing with Values

Global values are those that we can use in order to define our lives. There are many, but for the purposes of this discussion, we will use three - Work, Home and Health. Draw a chart containing these values on a sliding scale as seen below:

Values Scale

Work	Home	Health
Money	Family	Sanity

Examine the pairs of values and assign them a number from 1-3, with 1 the object/value you value most, 3 the one you value least. If you tend to follow the majority of the population (but don't be dismayed if you do not), the order that you arrived at is Home, Health and Work. Now circle the one you value the most (the one that received #1). Next, put a box around the one that gets the majority of your time/energy. Where did that box go? If you conform to the average, it was work. In order to support work as a value we must violate the things that we supposedly value the most.

Our primary value(s) are the one(s) that we support behaviorally. By behaviorally, I mean with time and energy. This situation where our primary value is not the one that we are supporting behaviorally is known as a 'values bind'. In order to support a primary behavioral value we will temporarily violate some or all of our secondary values. To support work, we violate home or health, we work late, we work weekends, we go home tired. In order to support home, we must spend less energy at work.

The best way to keep these values in perspective is to rotate your values. Make work a primary value, then shift that primary value to home, and you will find that you are now violating work to support home. Although it sounds easy, rotating values is not. Often our jobs themselves prevent them us from rotating values. Sometimes, it is our need for approval that prevents us from rotating our values. We tend to have a difficult time saying 'No' to people, or dealing with the disapproval we encounter from those to whom we say 'No' This constant need for approval at all levels invariably causes us to make bad decisions. Remember that if you try to make everybody happy, you will probably end up making no one happy - including yourself. We need to make decisions that support our values - cut your pie in such a way that we do this. Problem? Of course - we invariably let other people cut our pie for us. In the following sections, we'll examine how to prevent this from happening.

1. Salami Theory

Salami theory is not so much a theory as it is an alternate way of looking at a situation. Imagine, if you will, somebody hands you 2 pounds of salami, a loaf of Italian bread, a pound of Swiss cheese, and then asking you to turn these items into lunch. It is going to be a pretty difficult task without a knife, isn't it? In reality, all of our problems represent the salami, bread and cheese, each one a different problem of a different shape and size. Problems, as with the sandwich material, will be much easier to digest and swallow if we cut them up into smaller pieces and eat/solve them a bit at a time rather then trying to swallow them whole. Let's assume we have six problems that require semi-urgent attention (ok, in reality I realize that the real world has no such thing as the semi-urgent problem, but bear with me....). Since the human mind is not a true multi-tasking device, we are incapable of doing six things at the same time. We therefore require some type of time/priority management to accomplish this. Standard priority management courses tell you create A,B & C type tasks and accomplish them in that order.

My own experience of the last 8 months in a small DP shop (2 people) has taught me something different. One of the easiest ways to practise 'salami theory' is to take those six tasks I mentioned earlier, and break them up into pieces that allow you to solve them in the order that they are most important to the people who gave them to you. It is generally in your best interest to support those problems that will support you and perpetuate the 'illusion' of completeness. If this requires getting your projects to 95% and letting the rest take care of itself, so be it. I'm not proposing shirking your responsibilities, but be realistic and let other people help take care of some of the burden for you - ie: delegation. When writing that 20 page feasibility report, rather than spend time agonizing over the details, send out a rough draft, and let other people tear it apart/amend it at that point, rather than after you have put days of effort into it. I realize it always seems easier to write that 5 minute 'quick and dirty' program rather than spend the 2 hours required to teach somebody the basics of a 4GL that will allow them to do it themselves next time. A quick calculation should reveal though, that 2 hours spent in the beginning means that come the 25th request you are ahead of the game (2*60 minutes=120 minutes, divided by 5 = 24 requests, not to mention the goodwill that you gained in the process).

2. Dropping Problems that aren't Yours

Earlier in this paper I mentioned that we can't get people to solve problems that they don't own. The converse of this is we often take control/ownership of problems that have nothing to do with us, thereby creating additional stress for ourselves. Generally, it is due to our refusal to let go of problems once we have them. How do we end up with these problems? In much the same way that researchers catch monkeys in the wild. Researchers build a box consisting of bars narrow enough to allow the monkeys to get their hands through, then they fill the box with bananas. The monkeys see and grab the bananas, but the boxes are designed in such a way that they can't get the bananas out. The researchers then proceed to simply go and pick up the monkeys. Why don't they run? Because once the monkeys grab the banana they won't let go of them. If they dropped the banana, they would be free to run, but they persist in hanging on to it, and thus get captured. In most cases we get caught hanging onto problems in the same fashion.

When we are attaching ourselves to problems, there are two issues to consider: 1) do we identify with the values that are entwined with the problem? If yes, then take the problem on. 2) Is this problem in an area of 'unconsciousness' (reflex, instinct or guilt) for us? If yes, then drop this problem. If that means we have to stop our attempts to change a spouse, child or co-worker, then do it. The odds are always against you changing them anyway. This is especially true in the business world, as some experts believe that business is nothing more than group marriage. In business as in marriage, you can't get people to solve problems that aren't theirs (although we seem to often solve problems for them), and you can't enlighten the unconscious.

If you are tired of doing a co-workers job, one of those "Well, I've always been doing it, and somebody has to" situations, then stop doing it. Be prepared to see some consequences, both positive and negative. People won't be happy about the fact tasks/jobs aren't getting done, especially when the finger starts to point in the direction that it should have all along. Unfinished work takes very little time to be noticed, and as it is noticed things will change. If pressed for a reason, "I'm not doing it because I need not to do it anymore" should suffice provided you have sufficient grounds to stop performing the task. However, should you decide to keep doing the job and do nothing about it, then you will have to learn to stop complaining about it.

3. Maintaining Balance

Take a comfortable position, standing with your feet shoulder width apart, knees slightly bent. Imagine stress is about to push us over. The more balanced we are, the better prepared we are to cope with stress. The more balanced we are, the larger the force of stress required to get us off balance. However, when we are standing on one leg, very little stress is required to alter our balance. The difficult people we discussed earlier are often the people who manage to keep us off balance on a regular basis. Generally, difficult people are in good balance, and once they get us off balance, stress starts to affect us. This only creates more imbalance for us, and creates additional power for them. Our imbalance problems are often compounded when we get verbal as we start to lose balance. The discussion that turns into an argument, generally including the phrase, "I'm the boss here...". Don't worry, you are the boss, they know it and you should know it, but the fact you had to tell them should tell you something. Perhaps you aren't sure, or you wouldn't have told them. When we are off balance and verbal, we will often end up saying things we will only regret later.

Stress management is learning how to live with the situations when we are off balance until we can regain our feet. Our workplaces seldom change - even if our employers did us the favor of firing all of the 'jerks' and 'unconscious' people we feel we are forced to work with, it wouldn't help. They would only hire more to replace them. Perhaps the best comment I have ever heard about dealing with difficult people in the workplace came from my Grandfather, "The worst part about working is that you tend to spend your days dealing with little boys dressed up as men". We must learn to keep ourselves in balance, and if we can't, we must learn how to make good decisions when we are off balance.

A Dozen Quick Tips

As I started on this paper, a friend remarked that it probably wouldn't make any difference what I told people about stress management unless I could tell them that it would take effect in a hurry. As far as he was concerned we have fallen prey to the 'microwave mentality', we want it hot, we want it fast, and we want it now. Unfortunately, I have no 'microwave' solutions, hopefully this section will suffice.

As those of us with an interest in social psychology are prone to do, I spend a fair bit of time perusing bookstores and libraries checking what is available on the topic of stress/stress management in the workplace. The variety of information available is almost endless. While not professing to have read every available piece on stress in the workplace, I have read enough that I have compiled a dozen good tips (ones that have helped me) that should help you deal with stress on the job. While not every one will apply to each individual in all situations, I feel they offer some valuable suggestions.

1) Realistically appraise your current/future abilities. People sometimes overestimate their own abilities and expect too much from themselves and those around them. Always strive for your highest attainable goal, but learn not to resist it in vain.

2) Learn to laugh. Research has shown that when people laugh, it's often therapeutic. Work on developing a sense of humor and learn to laugh at yourself. Consider the following from Gerald Weinburg, author of "The Psychology of Computer Programming":

   "Last among the essential personality traits for programming, we might list a sense of humor. The computer doth make fools of us all, so that any fool without the ability to share a laugh on himself will not be able to tolerate programming for long. It has been said that the programmers national anthem is "Aaaaaaaaahhhhhhhhhh". When we finally see the light, we see once again that we have fallen into some oafish practise, or some truly witless blunder. It is only by singing the second verse, "Ha ha ha ha", can we long endure the role of the clown..."

3) Accept the things that are out of your control - traffic jams, an overbearing boss/subordinate, situations/people that are unlikely to change, regardless of what you do. Remember people cannot solve problems that are not theirs. Unless you can make their attitude/actions their own problems from their perspective, be prepared to live with it.

4) Don't procrastinate by doing low priority tasks because they are easy and fun - thereby neglecting high priority items until they become major crises. In short, practise good time/priority management.

5) Work at building supportive relationships both on and off the job. Friendships take time, but those with supportive relationships to fall back on often suffer less consequence under stress than those who are socially isolated.

6) Learn about the sources of stress (some of which you may have done by reading this paper). It is hard to accept the fact that stress sometimes comes from our own inability to handle life's disappointments.

7) Stressful events are in the eyes of the beholder and subject to the possibility of misinterpretation. Check your assumptions and perceptions to insure they are correct, and try to see the whole picture before reacting.

8) Avoid chronic "hurriedness". High stress personalities often try to do too many things too fast. Tell yourself that no enterprise ever failed because it was executed with caution and attention to detail. Learn to cultivate good judgement and decision making, not speed.

9) Learn to eat nutritious food. Unfortunately, it is easy to develop a habit of eating those things that satisfy our palate rather than those that satisfy the body. Even if this doesn't immediately reduce the stress levels you are under, your body will thank you for it.

10) When stuck in traffic or waiting in endless lineups use the time to be by yourself rather than fuming about the frustration. Visualize a more peaceful, tranquil situation.

11) Learn to feel comfortable talking about your problems. A close confidant is an invaluable resource during stressful situations. However,

you must avoid dumping on the some person all of the time. Another ground rule is that you must be available for those people in their time of need as well.

12) Allow for plenty of rest for mind and body. There is no substitute for rest and relaxation. In addition, those around you may find that the rest does wonders for your disposition, thus reducing their stress levels accordingly.

Summary

This paper was designed in order to present some basic outlines regarding stress management. As such, this paper may generate more questions than it answers. In order to help answer these questions, I have provided an outside reading list for those desiring more information. Hopefully this paper will pique your curiosity about stress management, and you can commence to create a reduced stress environment to work in. Stress management may be seen in some respects as self management, and we all owe it to ourselves (and those we care about) to take control of stress before it takes control of us.

Suggested Additional Reading

Adams, John	Understanding and Managing Stress San Diego, University Associates, 1980
Hanson, Peter	The Joy of Stress Toronto, Hanson Stress Mgmt. Org. 1987
Meichenbaum, Donald	Coping with Stress Toronto, John Wiley & Sons Ltd. 1983
Simmons, E.R.	Stress Control Proceedings: HP3000 IUG Anaheim 1984
Wills, James	Coping with Stress: A guide to living U.S.A., John Wiley & Sons Inc. 1982
Witkin-Lanoil, Georgia	The Female Stress Syndrome New York, Newmarket Press 1984

Systems Management and Operations Automation

Linda Hazzan
Carolian Systems
3397 American Drive #5
Mississauga, Ontario
Canada

1. Introduction

The computer industry today is driven by an aggressive pace of innovation, with faster, smaller processors being announced at an unprecedented rate. PCs are becoming more powerful and the classic distinctions between micro and mini-computers are being blurred. Single large computers no longer hold the bulk of corporate data, their role largely assumed by single user workstations.

Similarly, developments in data communications have had a great impact on the availability and distribution of information within the corporate structure. PCs networked to minis or mainframes allow users to manipulate corporate information directly, reducing the traditional dependence on large corporate data centers. These significant developments herald the beginning of what some are calling the new age of networked computing.

Today as always, technical innovations shape the way we work and the personnel structures within which we work. As traditional practices become outdated, efficient methods to rationalize and maximize the potential of the emerging technology must be developed. To meet this challenge, software and hardware vendors need to assimilate innovations into their product offerings.

1.1 Multi-Platform Networking

When choosing which hardware vendor to invest in, businesses must decide what kind of environment best suits their needs: centralized computing or distributed computing. Hewlett-Packard advocates the latter, and offers a wide range of computers to satisfy the needs of distributed environments. In particular, the HP3000 is the CPU most often found in companies which require distributed business applications. The ability to distribute hardware, and therefore applications and information, offers customers a cost-effective way to computerize at departmental and regional levels.

As organizations grow, so grows their demand for computers. Because of the wide variety of needs throughout a corporate structure, many organizations have chosen to use more than one hardware vendor in their distributed environment. Yet, since all these distributed machines are part of the same company, multi-platform interconnectivity which supports various operating systems and hardware architectures is also needed.

Even when they use only Hewlett-Packard hardware, some organizations use more than one operating system. Today, viable options for business applications in the HP marketplace are found under MPE V, MPE XL, HP-UX, and MS DOS operating systems. This is especially significant, since it broadens the scope of the business environment.

In recognition of the changes in the nature of distributed environments, hardware vendors and communications suppliers alike have realized the need for industry-wide standards to ensure that customers will be able to connect, maintain and control multi-platform environments.

As a step towards addressing this issue, the International Standards Organization (ISO) became involved in the development of industry-wide standards. The ISO proposed a seven-layer model, called Open Systems Interconnection (OSI), for data communications among various platforms. When such common protocols are in place, users have the ability to choose hardware and applications from a wider variety of vendors.

It is important to note that Hewlett-Packard is a strong proponent of the movement to implement and encourage standards within the industry. In fact, HP has adopted the OSI model and made it part of their range of communications offerings. This positioning by HP indicates their commitment to industry-wide standards that facilitate multi-vendor interconnection. To the customer, it offers more choice and flexibility for networking multiple systems.

1.2 PC to Mini Connectivity

Just as multi-vendor distributed applications are growing, so is the popularity of personal computers. PCs are quickly replacing terminals on the users' desks. Not only are they cost-effective as stand-alone machines, but their value is greatly increased when they are connected to corporate machines. The added value is basically two-fold: it allows for cooperative processing between the PC and the mini, and it offers a superior user interface to the host. Cooperative processing means that application processing can be split between a PC and the mini so that both machines are used effectively. To the developer, this means that applications can include both PCs and minis, allowing the different machines to handle that part of the application that they are best suited to. To the user, this means a consistent, intuitive interface that integrates multiple applications and standardizes simple interface procedures.

HP has recognized the flexibility and popularity of PCs in the user environment, and has incorporated PCs into their strategy. They have introduced "NewWave" as a standard for distributed applications among PCs and among PCs and minis. NewWave is based on Microsoft Windows and provides an integrated and consistent view of applications. It includes such features as support of a mouse, pull-down menus, and mechanisms for the easy implementation of on-line help. With its introduction of NewWave Office, HP has also introduced a new level of PC integration in the Cooperative Computing environment. NewWave Office offers E-Mail, document processing, and calendar and filing services to users. To further enhance the PC integration package, HP created VPLUS/Windows which allows PC users to run HP3000-based VPLUS OLTP applications from within NewWave. Perhaps the greatest impact NewWave Office will have on Cooperative Computing, however, is with its implementation of NewWave Agents. These agents give end users a window into the network, and allow the user to interact directly with the network and perform functions ordinarily funnelled through operations. (A more detailed discussion of agents and end user computing in the networked environment can be found later in this report.)

2. Network Management Standards

Certain hardware and telecommunications vendors have realized that it is not enough just to have standards for communications in place; vendors have to commit to multi-vendor network management so that control of all resources is also possible. As a sign of this commitment, eight major networking vendors created the OSI/Network Management (NM) Forum.

The OSI/NM Forum, as the name suggests, was created to help vendors implement standards for the management of networked environments. HP is an active member of the OSI/NM Forum and has committed itself to designing solutions that comply with industry standards. More importantly, this forum and HP have recognized the need for management of the entire networked environment and all its components.

The OSI/NM Forum has divided the concerns of users in networked environments into five categories. HP itself has added two more areas to this list, and has introduced a product called OpenView to address all seven topics. The categories are:

- **Fault Management:** The ability to identify, diagnose and resolve network problems quickly in order to maximize network uptime.

- **Configuration Management:** The tracking of network and device configurations easily and from a central location.

- **Inventory Management:** Knowledge of the size of a network and what all its components are.

- **Accounting Management:** The providing of information about computer usage for billing purposes.

- **Security Management:** Protection against unauthorized access, to make sure that only certain people get private information.

- **Performance Management:** Optimization of network performance through the collection and analysis of data about the network.

- **System Management:** The management of networked systems from a central point, for consistency and to reduce staffing costs.

HP is offering OpenView as the integrated network management system for multi-platform users. At present, it is mostly a shell product which will allow modules to be added to it so that each of the seven user categories are addressed. Graphic User Interfaces to OpenView will allow users to configure, control and manage their entire network from a single management station.

2.1 Network Systems Management

Carolian Systems has also recognized the need for management tools in a networked environment. Of the seven areas determined by HP, it is the seventh category, System Management, that this paper will address. Executives in MIS want an overall perspective of their network. There are often many dependencies among machines in a network, so the independent management of each HP3000 is an inefficient way to control the networked environment. Consolidated control and management of multiple HP3000s would allow all information to be easily related. The result is that MIS staff can make better decisions and increase overall network uptime because they are working in a more controlled environment.

The issue of managing all aspects of networked systems from a single point is parallel to the need for managing all components of a network (both communications and systems). Just as there are seven definite issues that HP has chosen to address with OpenView, there are specific areas that customers need to manage within the area of network systems management.

The topic of network system management includes such issues as:

- **Performance Monitoring:** The ability to spot bottlenecks which slow user response time.

- **Capacity Planning:** The ability to mange current resources and forecast future requirements on a network-wide basis.

- **Spooling Management/Report Distribution:** The flexibility to automate and expand the capabilities of report distribution with a centralized perspective.

- **Batch Processing Management:** An integrated monitor and scheduler that encompasses all jobs on all systems.

- **Production Schedule Management:** The ability to plan and control production schedules for an entire network.

- **System Trouble-Shooting and Failure Recovery:** The ability to define and detect critical events, and to respond either automatically or interactively in a timely manner. This also includes the ability to supply technical experts with the information necessary to resolve problems from a central location.

- **Regular System Maintenance:** The ability to install, maintain and update software; to configure systems from one location; and to do back-ups of all data from all systems automatically and remotely.

- **Security Management:** To prevent unauthorized access to any and all systems in the network and to report on attempted violations.

- **Database Management:** The ability to monitor and control database structure and size from a central location.

These represent the foremost issues that system managers need to address in any HP3000 environment. In the case of multiple machines, the consolidation of information, integrated monitoring and control, and automation become critical if management is to have an overall view of company resources. The only way for these objectives to be achieved is through the use of system management utilities.

2.2 System Management Utilities Today

System management utilities are software and hardware products which address the different system management areas outlined above. They have long been used by system managers trying to streamline operations to provide better service to their users. These utility systems tend to supplement operating system services where these are found to be wanting.

Typically, system management utilities have one or more server processes which perform much of the system-related processing required. A batch scheduler, for example, has a job monitor process which automatically executes the production schedule and verifies the successful execution of specific jobs. Much of the technical complexity of system utilities is to be found in this module.

The server process usually runs under a permanent job which must be active for the system to perform its assigned functions. The process will detect particular system conditions and automatically react to them in some pre-determined way, usually logging its progress. Additionally, the server will almost always accept on-line requests for status information, temporary configuration changes and specific system-related tasks, which it then executes.

The background processing which servers perform is, in most cases, controlled in two ways. The first is through a system administrator's interface. With this interface, the system manager pre-configures the operating parameters of the server processes and performs any maintenance functions the server might require. The second controlling mechanism can be thought of as an operator's interface. This interface usually allows the on-line requests for status and specific system tasks described above.

We can therefore identify the following components in any given system utility:

1. Utility server processes
2. Configuration and maintenance user interfaces
3. Monitor and control user interfaces

These may be implemented in a variety of ways, depending on the requirements of the system being designed. All user interfaces may be implemented as a single program having a dedicated server module. On the other hand, a given utility may have distinct interfaces for operator and administrator functions, interfaces which share unique system servers. But regardless of implementation, these three components will invariably appear in one form or another as part of utility products.

How these components are designed determines to a large extent the adaptability and modularity of the utility system. Until recently, most system utilities were designed to deal with the processing of a single computer and took no advantage of networked environments. Each computer would maintain its own copy of the utility, together with private configuration information. Little or no dialogue occurred between the utilities running on different systems.

Today, the proliferation of networked systems exerts significant pressure on developers to adapt these single machine utilities to a networking environment. Indeed, much progress has been made in the area of networked systems integration in the HP3000 utility marketplace. But while some products have implemented multi-machine mechanisms, most notably batch schedulers, these mechanisms integrate only small sections of the systems, leaving many of the operational and managerial issues untouched.

Typically, integration has occurred at the utility server level, but little progress has been made at the user interface front. Here, operational and managerial staff is presented with a highly fragmented view of computer resources. From the perspective of the system manager, having to deal with isolated systems complicates the process of monitoring, controlling, configuring and maintaining those machines. Without the consolidation of data, MIS has to deal with vast quantities of information that need to be analyzed for all systems. Today, much of this collection and analysis is done manually, or not at all.

The problem is magnified if the machines are geographically dispersed. In this case, many companies are faced with the question of how to support machines remotely. One solution today is to hire staff for those remote locations. This is an expensive way to manage systems, and still does not satisfy the need for a network-wide management view or control centre. Without integrated interfaces, the logistics of adequately servicing networked systems may prove insurmountable.

Not only is it very difficult to effectively service networked systems, but it is also hard to implement network-wide policies. Often organizations would like to have standard procedures for operations and standard methods for problem solving. However, in today's distributed set-ups, this kind of standardization is difficult because information cannot be easily shared, nor is it consolidated anywhere. There may be much duplication of effort among sites or, if policies are in place, they may not be suitable for all machines. Once again, management needs an overall picture of data processing.

The difficulties only increase when the issue of multi-platform networks is introduced. These are becoming widespread at a significant rate, and the demand for systems which can easily accommodate migration to new platforms will be strong. Advances in the area of multi-vendor connectivity and international communications standards mean that installations are closer every day to implementing heterogeneous networks in which system management tools will hardly resemble each other.

As a result, the state of the art in system utility software is not prepared to deal with the challenges presented by modern networked environments and forces users into inefficient procedures for managing multiple machines. Whatever gains developers make in increasing the power of networking, are lost when customers discover we offer no means of efficiently managing networked systems.

3. Moving Forward into the 1990's

To respond to the changes and trends in the marketplace, and the focus on multiple systems environments, HP3000 users are looking for a way to manage and control networked computer systems through interfaces that are designed for the specific tasks to be performed. The following section outlines how the roles of individual MIS staff members may change, and how the interfaces to the information they require also need to evolve. This rationale for reorganizing information and how to access it will outline a new architecture for the management and automation of systems.

3.1 Concepts

In order to propose an architecture, we first need to understand the general principles and requirements which come into play when dealing with networked systems. Clearly, the issues networked system managers will deal with are much broader in scope than those related to isolated computers. On the other hand, a responsive utility environment will not prevent users from focusing on specific details. We therefore need to maintain a careful balance between global issues on the one hand, and the ability to control and monitor individual system components on the other.

A successful implementation of networked systems utilities will require integration at two levels. At the first level, utility servers must implement sophisticated network-wide control over their domains. For example, batch processing servers must support network-wide dependencies and be able to launch jobs on remote systems, and information about the success or failure of jobs executing throughout the network must be easily obtainable. In addition, status information from different utility servers must be consolidated in a rational way. For example, it only makes sense to relate the successful completion of a job to the successful distribution and printing of any reports that the job generated.

To accomplish this level of integration, a network-wide utility monitor which controls all servers and gathers status information from them is required. User interface processes have access to this information and control, through their connection to the utility monitor.

A necessary second level of integration involves the interfaces to the utility systems. To specify the requirements these interfaces must meet we should first understand the different functions and responsibilities systems management staff must fulfil.

We have identified four distinct kinds of personnel with different needs for access to information and control: The Networked Systems Operator, the Networked Systems Manager, the Networked Systems Specialist, and the Networked Systems End User.

The Networked Systems Operator

The operator is primarily concerned with the execution of the production schedule and is responsible for its successful and timely completion. Depending on the environment, the operator may also verify that reports are distributed to remote printers and that interactive users receive adequate service. In addition, the operator must fulfil any requirements a given processing system might have for human intervention.

Bearing this in mind, we can see that the operator needs highly dynamic status information and control access. He or she must be notified of problems and system requests quickly and must be provided with the means to isolate errors and correct them. In an automated shop which runs a batch scheduler and a report distribution spooler, the job of the operator is largely one of monitoring and error recovery.

To meet the requirements of a Networked Systems Operator's interface, the monitor and control functions of all operations utilities need to be under a single module which provides highly integrated network-wide access. This presents a global view of production as well as mechanisms for controlling components individually. Through this interface, operators can easily monitor the status of systems throughout the network, can zoom in for a closer look to isolate problems, and finally can alter the processing of specific system components. (See Figure 1 for an example of the Networked Systems Operator Interface.)

To be successful, the interface must incorporate a priority-based alert system and fast online retrieval of all information required to solve a potential problem. If a critical job aborts,

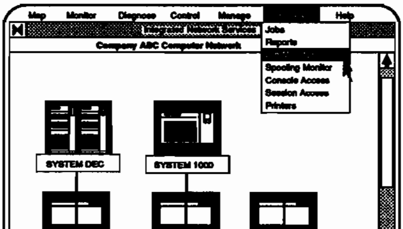


Figure #1

The Networked Systems Operator's interface combines the monitor and control functions of all operations utilities for example, the notification to the operator should come with an information package which includes the reason for the failure and instructions from the system manager on how to recover the job.

The Networked Systems Manager

The system manager is responsible for implementing procedures and schedules for production. The job largely consists of configuration and maintenance tasks. The manager must ensure that computer resources are used effectively and that batch job throughput and response time of interactive processes are both within acceptable levels. He or she may also undertake capacity planning and performance tuning responsibilities.

The needs of the system manager therefore fall into two basic categories: pre-production planning and post-production evaluation. The manager needs to configure the utility systems and instruct operators on procedures before production, and then examine the efficiency of the operation after production.

The Networked Systems Manager interface must provide the ability to configure and maintain all utility sub-systems from a single integrated interface. The manager will only have to configure a piece of information once for all utilities to have access to it. This is all the more important in an environment where the different utility modules cooperate to create a unified view of processing. The system manager is relieved of the requirement of creating a large number of distinct configurations which must nevertheless conform to each other in order for the whole system to function correctly. The intent is to present the utilities as part of a single system which can be configured as a unit.

Another important requirement is to allow system managers to quickly and easily establish policies and provide instructions for operators to work with. Post-production reports need to be organized so as to inform the manager of the processing accomplished, any failures which occurred, the reason for those failures, and what operators did to correct them. The manager can then use this information to create on-line help for operators. This help will be delivered to operators next time they receive an alert to a given problem, so when a particular production job aborts the operator will have all the expertise of the system manager a few keystrokes away. He or she can stream a recovery job to expand database capacities, alter the operating parameters of certain application processes, or further investigate areas the system manager has identified as possible causes for the failure.

In turn, the system manager can learn more about his or her operations by reviewing post-production reports. This iterative process of creating operator documentation guarantees that in time a valuable knowledge base of operational procedures will be built up, and expertise accumulated by operations staff can be shared, even after that staff has left.

In addition, the system manager can configure keyword-driven on-line help to document procedures and regulations operators must follow during production. For example, he or she can document the steps required to take a full backup or run month-end financial reports.

The Networked Systems Specialist

The Networked Systems Specialist is assigned a specific area of processing as his or her main responsibility; for example, system performance tuning or the efficient operation of a particular application. This individual will therefore want to be intimately familiar with the status and performance of his or her application domain throughout the network. To support the systems specialist, a means to highlight information relevant to the assigned area and filter out data about other systems must be provided. Certain events which may have a high priority for operators only distract the specialist. Conversely, having specialists means that operators can get on with the production schedule without worrying about areas best handled by experts.

Because of this special focus, the specialist requires a much greater level of detail in status information than operators. After all, system-related trouble-shooting is this individual's greatest talent. The Networked Systems Specialist interface must support the same kinds of services offered to both system operators and managers, but must be geared towards a specific area of focus. Here too a careful balance between specifics and summaries must be struck.

The Networked Systems End User

The Networked Systems End User is any individual requiring access to information in the network. Through the graphic user interface, the end user would use task automation services, called "agents", to interact directly with the network servers and perform routine batch-type operations. For example, a user could run a report in batch directly from his or her workstation simply by clicking on an agent icon. This request would then be sent transparently to a network server. A network spooler server integrated with the end user's interface program would pick up the report and automatically route it to the printer closest to the user, then tell the user which printer the report was sent to.

This level of PC integration is a key thrust for data center management and automation solutions of the 90's. Referred to as Personal Operations Management, this direct interaction with the network allows organizations to minimize the workload of their operations staff by handing over tasks most logically performed by the users themselves.

3.2 Features

Networked Systems Integration

The single most important feature of a control and automation environment is its ability to provide system users with an integrated view of their networked operations. As discussed, this integration is carried out at both the functional level of the utilities and at the user interface level. The result is that all utilities are presented as modules of a single system with a network-wide scope.

PC-based User Interfaces

Unquestionably, the best method of delivering this integrated solution to the user is through PC-based interfaces. By standardizing on user interfaces, a consistent look and feel is offered in addition to all the ease of use that such state of the art "windows" technology brings with it.

The advent of sophisticated PC-to-mini connectivity allows interfaces to communicate with integrated utility servers, and give users a window into the network so they can communicate with the integrated services directly.

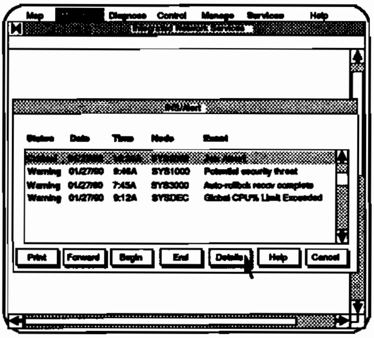
Priority-based Alert Functions

Alert functions are simply the efficient distribution of status information, usually signalling a high priority event that requires human attention. Because of this, alert functions are best implemented as a series of reporting options which can be employed depending on the priority of the event signalled. For instance, system failures are almost always very high priority events. Because of this, they should be delivered to the Networked Systems Operator interface for immediate action. Other events are best bundled together and regularly delivered to a specific individual via electronic mail.

In the Operator's interface, for example, alert messages can be displayed on an alert window in various colours to reflect levels of urgency. These messages will not scroll off the screen until the operator indicates they have been addressed. This ensures that high priority events are given the required attention. On the other hand, status messages which do not require any human intervention are displayed once and allowed to scroll with other text.

The machine window on the operator's interface also uses colour to indicate the urgency of the highest priority alert outstanding for a given system. The operator can quickly get a feel for the state of all machines in the network with one glance at this screen.

Automation and management solutions should also allow alerts to be delivered by phone through a digitized voice or paging system. This unique option means that operators need not be tied to a system console. They will be notified of urgent alerts whether they are in the computer room checking the production schedule, in the finance department helping a beginner get used to a new application, or even at home after hours. Having received an alert, the operator need only reach the nearest workstation to run an Operator's interface and address the problem.



Network-Wide System Manager's Interface

This interface offers the kinds of services any system manager will appreciate in a networked environment. Through it, the system manager can generate post-production reports as well as issue on-line queries on past production. The interface also allows the manager to maintain a network-wide configuration of the entire environment from a single management station. Facilities for creating the on-line help sub-system for operators should also be included. Another important feature of such an interface is that it should offer the ability to automatically install and update software throughout a network.

4. Summary

The landscape of business computing is changing at a rapid pace. The increasing technical and financial feasibility of large computer networks is promoting a distributed processing strategy as a cost effective way of implementing enterprise-wide automation. As a result, the number of functional networks connecting a wide variety of computer equipment throughout organizations is growing steadily. This growth should continue and indeed accelerate as the cost of technology decreases over time.

These large corporate networks present new challenges to operators and systems maintenance staff charged with managing the computer systems. The difficulties involved in effectively monitoring and controlling multiple machines are significant, especially when those machines are geographically dispersed. The cost of staffing such networks is growing dramatically, to the point where it far outweighs the cost of computer and communications hardware. In some cases, maintaining a network of machines is expensive enough to render the whole implementation ineffective.

Today, a key component is missing from communications solutions in the computer industry. While hardware vendors provide the means to interconnect distributed processing systems on a large scale, they offer little means of efficiently managing those systems.

What is required is a sophisticated networked systems management and automation solution to fill this void in communications technology which will provide managers with an integrated utility environment for controlling networked computer systems from focal point stations.

By addressing this need and therefore, enhancing the performance of networks with multi-platform systems, third party vendors such as Carolian Systems will help to ensure that companies receive maximum return on their computer investment.

References: Hewlett-Packard's OpenView by Lance Ketterer Cupertino, CA

Controlling External Devices Through Terminal Ports

By Grant Parry
Summit Information Systems Corporation
P.O. Box 3003
Corvallis, OR 97339

Computer Museum

Several strategies can be used when faced with the task of communicating with an RS-232 device. The important decisions are which type of port to use and the type of flow control to employ. Knowledge of HP ports plus some trial and error with the device in question should usually lead to a logical strategy from the wide range of possibilities.

Data Transfer

The first concept to understand when communicating with an external device is data transfer. Data transfer describes the actual flow of data, transmitted in electronic pulses from one device to another.

In order for any two devices to communicate, there must be a cable between them. Cables are designed according to RS-232 specifications; the manual for your external device should list its cable requirements. One thing to remember is that the HP 3000 is considered Data Computer Equipment (DCE) in RS-232 terms and if the peripheral expects to connect to Data Terminal Equipment (DTE) you will need to modify the cable.

When you connect DCE to DTE, Send Data on the DCE equipment is the same pin as Receive Data on the DTE. If you need to connect DCE to DCE, you must fool the two devices by "rolling" the pins on one end (instead of pin 2 connecting to pin 2, in a rolled cable, pin 2 connects to pin 3). Common pins to roll are: 2 & 3 (Send and Receive Data), 4 & 5 (Request/Clear To Send) and 6 & 20 (Data Set Ready/Data Terminal Ready).

Several rules govern data transfer. The speed of the transfer in pulses per second is known as the baud rate. The number of pulses per character is the data length and is usually seven or eight bits. In addition, every character sent is framed by a start bit and one or two stop bits. Data may be transferred in either synchronous or asynchronous modes. The character set being used (such as ASCII) provides a lookup table between display characters and the pattern of pulses which represent them. Also, there may be a parity setting, which serves as an additional indication that the data transfer is synchronized.

All data transfer specifications must be compatible for two devices to communicate. HP 3000 specifications are that it is capable of operating at most standard baud rates, with seven or eight data bits. Each character is framed with one start bit and one stop bit. It can only use the ASCII character set and uses asynchronous communications.

In addition to these specifications, there is also the issue of full duplex or half duplex. This is a fancy way of saying whether the device on the other end is capable of echoing the data sent by the first device. In full duplex, each data character sent by one device is returned to the other. This is the way that the HP port behaves when a terminal is used. The data entered is actually sent to the HP and back to the terminal before it is displayed on the screen. Half duplex provides no echo of data sent. Half duplex is probably the mode you will want to use so that data sent by the peripheral will not be echoed back by the HP port controller. This can be accomplished by the use of FCONTROL, mode 13. (See the MPE Intrinsics Manual.)

Flow Control

Once the two devices are talking, flow control will have to be addressed (unless you are only sending a small amount of data between devices). Flow control is the ability of one device to say "hey, wait a minute" when it has received too much data and is in danger of losing or mishandling additional data if it comes too fast.

The two major classifications of flow control are hardware- and software-based. Hardware-based flow control means that there are additional wires in the cable between the two devices that allow them to affect each other. Software-based flow control means that flow control commands are sent in the data stream. The devices read the data and strip the flow control commands as they come in.

Request to Send/Clear to Send (RTS/CTS) is the most common form of hardware flow control. What this means is that when one device is ready to send data to the other, it will provide voltage on the Request to Send pin. If the receiving device is ready, it will provide voltage to Clear to Send pin.

Software flow control is divided into two major types; XON/XOFF and ENQ/ACK. XON/XOFF is an assertive flow control--the device which wants a pause in data transfer will send an XOFF to the remote device. When it is ready for more data, it will send an XON. ENQ/ACK is a passive flow control--the sending device will initiate the flow control by sending an ENQ. If the receiving device is ready for more data, it sends an ACK and the sender transmits data. If it does not return an ACK, the sender waits for the ACK before transmitting more data.

Types of Ports

The distinction between ports on the HP 3000 are whether they are terminal or modem ports. Terminal ports are simple; they use a three-wire cable (Send Data, Receive Data and Ground). Only XON/XOFF or ENQ/ACK flow control may be used (on MPE XL, ENQ/ACK is no longer an option). Modem ports are more complex; they use a 12-wire cable and are able to provide the hardware level flow control required by modems.
Because they are simpler (and because there are more of them) terminal ports should be used when possible. If the peripheral device being used allows a choice between CTS/RTS and XON/XOFF flow controls, XON/XOFF should be selected for a terminal port. If both CTS/RTS and XON/XOFF flow control are required, a jumper between pins 4 and 5 on the peripheral end of the cable will automatically put voltage on CTS when the peripheral sends RTS. Make sure the device can XON/XOFF before doing this. If not, RTS/CTS and a modem port must be used.

The MPE File System

All of the actions taken by your program with regard to the port will be performed by the MPE file system. The file system provides eight intrinsics which relate to the use of terminal ports. These intrinsics are FOPEN, FCLOSE, FREAD, FWRITE, FCONTROL, FCHECK, FSETMODE and FDEVICECONTROL.

Your program will start communication with the port by calling FOPEN and passing the LDEV number in a string as the DEVICE parameter of FOPEN:

```
MOVE DEVICE:="30";
FNUM:=
     FOPEN(FNAME    << FILENAME BA >>
          ,%604 << FOPTIONS LV>>
          ,%504 << AOPTIONS LV>>
          ,128   <<RECSIZE IV >>
          ,DEVICE     << DEVICE BA>>
          );
```

Once you have opened the port, you will want to condition it with FCONTROL and possibly FDEVICECONTROL. FCHECK can be used to determine the exact nature of any problems which occur during any operations on the port.

Writing data to a port is just like writing data to any other file. The file number received from FOPEN is passed to FWRITE, along with the array containing the data to be transferred and the transfer count (the number of bytes or words).

Reading from the port is a little trickier. Just because a wakeup sequence has been sent to the peripheral does not guarantee it will respond. To avoid having to wait forever for the response, a timeout should be set on the port using FCONTROL, mode 4:

```
TIMEOUT := 60;  <<60 second limit>>
FCONTROL(PORT'FNUM,4,TIMEOUT);
IF <> THEN BEGIN
     ERROR := 11;  <<FCONTROL failed>>
     FCHECK(PORT'FNUM,FSERR);
     END
```

Also, because the FREAD to read the data has the potential of failing, the condition code should be checked to see if the read failed and FCHECK called for more details in the event of a failure.

Three intrinsics are provided to condition the port. FSETMODE allows the automatic carriage return, line feed sent by the file system on every line to be inhibited; FDEVICECONTROL allows the workstation configurator to be altered; and FCONTROL can be used for a variety of control purposes.

Reading Data Through the Terminal Port

In order to use most peripherals, bidirectional communication will almost certainly be necessary. From the point of view of the programmer wishing to read or write data on the terminal port, the task is fairly simple--once the port has been allocated with FOPEN, call either FREAD or FWRITE, depending on the operation desired.

If the device being communicated with is compatible with the way the HP 3000 handles flow control, there will not be any problems. However, most devices do not act like a user with a terminal.

Let's take a look at "standard" terminal port input/output. Starting with the program which wants to read data into the port, the FREAD intrinsic is called. Eventually, the request for the read makes its way to the port controller for the terminal port in use.

The port controller begins the read by initializing the terminal buffer. Initialization flushes the buffer which means any data which was received into the HP 3000 port before the read was posted by the program is lost. Once the port is initialized and the port controller is ready to process incoming data, an XON character (DC1) is sent to the remote device, signalling that the HP 3000 is ready to read data.

The HP 3000 will continue to read data until either a carriage return is received or the number of characters received reaches the number expected by the program.

No XOFF from the HP

So far everything seems right; the HP 3000 is acting just like it is supposed to, sending an XON (DC1) character when it is ready to read data. Unfortunately, the HP 3000 does not send an XOFF (DC3) character when the terminal read completes. Most of the time, the way the HP terminal port works does not really matter, since it has very little effect on usage by individuals with terminals; they don't see the XON in the first place, and since the system normally runs in full duplex (echoing all characters sent by the terminal back to the port), most users just continue to type until the data they are typing is no longer being echoed (the port isn't listening to the device).

Most peripheral devices, however, want to run in half duplex; even when running full duplex they do not have the human judgement to decide that the HP 3000 is no longer accepting data. This means is that if you are communicating with an external device which relies completely on device-level flow control in the form of XON/XOFF, you're in trouble because the HP 3000 never gives the XOFF and the device continues to send data. Remember what was said about the way the port controller starts a read? By flushing the contents of the terminal buffer at the beginning of the read, the HP 3000 port controller assures that any data sent by the device between the program's FREADs is completely and irrecoverably lost.

The only way to avoid a loss of data due to flow control is to make sure that the device pauses at the end of each data record to wait for the HP 3000 to post the next read (signalled by the DC1 coming to the device). Since the HP 3000 port will continue to read data until either the end of line character is received or until the character count specified by the program has been met, you will probably want to post large reads on the port and expect the receipt of a carriage return to terminate the read on the HP 3000. Once a carriage return has been sent , the device needs to wait for an XON (DC1) before it continues.

Special Port Conditioning

It may be possible that the peripheral does not terminate the data it sends with a carriage return and will not be sending a constant number of characters. If FREAD is called on a port in normal mode, this can cause the port to hang forever (or to timeout if one was set). If the final character of data sent by the device is constant, the read can be terminated by using FCONTROL, mode 25 to control the line termination characters for terminal input:

```
TERMINATORS.(0:8):=0; <R>
        <<Sets CNTL-Y to null character>>
TERMINATORS.(8:8):=CR;<R>
        <<Sets RETURN to RETURN>>
FCONTROL(PORT'FNUM,41,TERMINATORS);
```

```
IF < > THEN BEGIN
      ERROR := 12;
      FCHECK(PORT'FNUM,FSERR);
      END
```

You may also be faced with the problem of sending data to the device and having it respond before the read can be posted on the port. This results in loss of data sent before the read. In this case FDEVICECONTROL can be used to control the character generated by the HP 3000 (usually XON) when it is ready to read data:

```
LOGICAL ARRAY LCLBUF(0:0);
BYTE ARRAY LCLBUF'(*)=LCLBUF;
LCLBUF := 0;
<<into LCLBUF, we put the ASCII number>>
<<of the character we want, in this case>>
<<we want a line feed, so we pass 10 >>
LCLBUF'(1) := 10;
FDEVICECONTROL(PORT'FNUM,LCLBUF,1,192,32,3,FSERR);
IF < > THEN BEGIN
      ERROR := 13;
      END
```

Binary Data Transfer

Binary data transfer could be another potential obstacle when transferring data between an external device and an HP 3000. A carriage return (and optionally a control-y) will terminate the read in progress on the port. Normally, this is desirable; however in the case of binary data transfer you never know if a carriage return is a bona fide end-of-record indication or if it is just a random combination in the binary data. Or suppose you want the carriage return to be given to your program as part of the records instead of terminating the read in progress? FCONTROL allows certain default settings and operations on terminal ports to be overridden. To enable binary data transfers, use FCONTROL, mode 27.

Binary data transfer means that the HP 3000 will become completely transparent to any of the data that is received even if it is a control character or a carriage return. This allows carriage returns to be received into the data stack of a program. But remember, port reads do not get terminated by a carriage return anymore. The read request can only be terminated by meeting the character count.

The first impulse when encountering this problem is to put a timeout on the port. This way, when the read request by the HP 3000 is not satisfied, eventually the timer governing the maximum duration of the read is exhausted and the read is cancelled by the system. Unfortunately, the port timeout works differently-- once the timer has exhausted, the read is terminated but the HP 3000 does not return the data which did come into the port to your program. Therefore you can only use the port timeout in situations where you do not need to know what was in the partial record received before the read timer expired.

Reading binary data into terminal ports is no small task and requires a sophisticated solution. Generally speaking, you will need the cooperation of the maker of the peripheral you want to interface to. Failing that, you should be able to find some special purpose hardware buffers which overcome the way in which the HP 3000 handles flow control and record termination.

Writing Data Through the HP Port

Although the majority of this discussion has dealt with the flow of inbound data from the HP 3000, there is a gremlin of HP terminal output which deserves attention. The HP 3000 is generally quite adequate at stopping and starting its transmission of data upon receipt of XOFF/XON, with one notable exception--a situation which will cause the HP port to hang, waiting for the receipt of an XON which will never come.

The problem occurs when the program is transmitting data through the HP port at a rate which the peripheral cannot keep up with. The peripheral begins to send XOFF/XON pairs to the HP port as it gets behind and then catches up with the flow of data from the HP.

Meanwhile, the program, which is not directly controlling the HP port but is rather passing requests to the port controller, continues to execute and gets farther and farther ahead of the controller (which is being impeded by each XOFF from the port).

Eventually, the program has one or more read requests which have been queued for the port and added to the terminal buffer, but have not yet been sent. Now, let's say that the program has reached the end of the data to be transferred and is going to read the port, expecting data in the form of a response from the peripheral.

Meantime, the controller has been busily writing data to the port. Somehow the read requested by the program is posted to the port in front of the data being written. The controller at this time is in an XOFF state waiting for an XON to come back to it so it can finish the write. But the XON character from the peripheral has been interpreted as data and passed to program as part of its out-of-sequence read request.

Now, let's stop and take a snapshot of where everyone is. The program is in the middle of a read request. An XON character (from the peripheral) has been placed in its buffer. The controller on the HP is in an XOFF state waiting for an XON from the peripheral (the one which was sent to the program instead of being treated as flow control). It will continue to wait until an XON character is received from the peripheral. The peripheral, however, has sent an XON and is waiting for the rest of the data to be received from the HP. It will continue to wait until it receives the end of data indication from the program (the FWRITE which has been requested, but not serviced by the controller).

Everything is stuck until someone either issues an ABORTIO command for the LDEV of the port, aborts the program or resets the device. Also, the hang could be cleared if the peripheral had set a timeout and sent another XON to the HP port if the timeout occurred.

If you are experiencing this situation, there are a couple of alternatives available which should rectify the problem. The first choice is to transmit some additional data (such as NUL characters) to the peripheral which will not be likely to cause it to send back an XOFF. That way, the XOFF/XON isn't happening right as the program gets to its last write and starts to read.

Another alternative which may be easier to implement is to introduce some software pacing, in the form of using the PAUSE intrinsic between each few writes. In this case, it is often beneficial to find out the size of the receiving device's communications buffer and under what conditions it puts out an XOFF. By

PAUSEing at strategic intervals, you can give the device time to process all of the data you are giving it, without it having to send XOFF characters.

Conclusion

Interfacing external devices to HP ports can be tricky business, but if you are willing to endure some trial and error, it can be very rewarding. The best way for you to build a reliable interface is to watch the flow of data between the device and the HP. That way, you'll be able to get a good feel for what is happening and what could go wrong with the interface.

It's also a good idea to try to have a sample dialogue with the peripheral and a terminal before attempting to communicate with it through the HP port from a program. The peripheral manual should give you the proper cable configurations to do this, just remember that HP ports are considered DCE and as such will be the opposite of terminals, which are classified as DTE.

Acknowledgements: I would like to thank Ross Scroggs and Randy Medd of Telamon for their valuable assistance, advice and tutoring in the wondrous ways of HP ports.

DISK ARRAYS - MASS STORAGE OF THE FUTURE?

Ed Pavlinik
Hewlett-Packard
Disk Storage Systems Division
11413 Chinden Boulevard
Boise, Idaho 83707

INTRODUCTION

Disk storage arrays have recently become a leading topic of discussion in the computer systems business, not just among industry gurus, but also at trade shows and user group meetings worldwide. A variety of papers have been published and numerous articles have appeared in industry publications. As a result, innovative users are more than mildly curious about the ramifications of using these devices on their computer systems. In fact, some users may already be using disk arrays on PC based systems.

What makes the disk array different from traditional on-line disk storage devices and how can the typical data processing manager take advantage of this new technology? The goal of this paper is to provide an understanding of disk arrays as a potential solution to changing mass storage requirements.

THE DISK ARRAY CONCEPT

Large Capacity Storage from Small Disks

The typical disk storage array is basically a mass storage system utilizing a large number of small form factor disk drives, such as 5.25" or 3.5", which are linked together with an intelligent controller to provide a large amount of disk storage. Capacities of the 5.25" diameter disks have been steadily increasing as disk manufacturers push the limits of magnetic recording technology. Similarly, 3.5" diameter disk drives have also been increasing in capacity due to improvements in track and bit densities.

Since the smaller diameter disks are also being mass produced in large volumes, production economies of scale will result in lower manufacturing overhead and cost per megabyte when compared to eight-inch or fourteen-inch disk mechanisms. Obviously, the same storage capacity can be designed into a large single spindle high capacity disk drive. The drawbacks of the single large expensive disk are higher manufacturing costs due to lower production volumes and potentially lower performance, since so much data is stored on a single spindle under one actuator.

The disk storage array allows larger computer systems to take advantage of many of the benefits associated with smaller form factor disk drives and overcomes some of their limitations through the intelligence built into the disk array controller. The disk array controller can be designed to offer a high degree of flexibility in meeting diverse user requirements. Optimization of the various tradeoffs can also be achieved in a general purpose array or if appropriate, other special arrays can be designed to solve specific user needs.

Array Terminology

The terminology used to describe disk arrays can lead to some confusion, since several vendors use the term "disk array" to describe different products. For example, some data sheets refer to products as disk arrays, even though technically they might more properly be described as conventional disk storage systems or disk cluster controllers. By another definition, the term disk array consists of multiple disk drives under the overall command of a single controller. Despite possible confusion in terminology, the primary function of the array is to increase total storage capacity as well as to improve performance by selectively spreading files over multiple disks. The file spreading technique utilized by the intelligent controller is called disk striping. Striping is a technique which writes a single byte of information to each disk drive in the array in a parallel fashion. All the striped disks work in unison on a single I/O transfer.

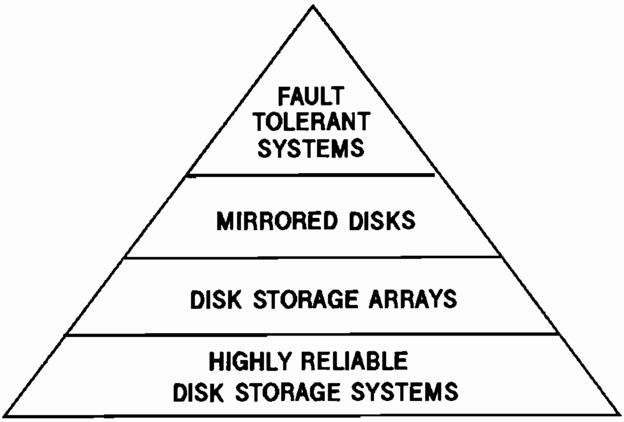
Disk striping can also be accomplished in software by designing a special disk driver to spread files across several individual disk volumes. In this case the computer performs all of the data manipulation as well as the disk management and as a result, the overhead may be quite high. In a disk array the array controller does the striping outboard of the CPU, thus saving CPU cycles for other tasks. Products are available from a number of vendors which illustrate both methods of disk drive data striping.

Data Availability Hierarchy

A good way to visualize the concept of data availability is to consider a complete hierarchy of data protection. In this model, higher data availability is achievable via various techniques for a corresponding increase in cost. Depending on the level of availability desired, one can pick a point on the cost curve for the specific solution desired. For example, if a medium level of availability is perfectly adequate for some applications, there are solutions available to meet those needs for a lower cost than the completely redundant fault tolerant solution.

Here's an example showing these various levels of availability starting from highly reliable disk storage systems and then working upward to progressively greater data availability through the use of such innovations like disk arrays, disk mirroring, and fault tolerant systems. As you might expect, costs will increase as one moves toward the top of the pyramid. Many applications, however, demand these high availability systems despite the higher costs, since the benefits outweigh them. It all depends on the cost of system downtime unique to every business.

DATA AVAILABILITY HIERARCHY



The highest level in this hierarchy represents fault tolerant systems which incorporate full redundancy and include disk storage systems with no single point of failure. As a result, a system is created that features almost no downtime. These systems represent perhaps the ultimate in high data availability for use in mission critical applications and offer continuous data processing with such features as multiple processors, redundant I/O, and software checkpoints.

Disk drive reliability may be considered to be a cornerstone or foundation in this hierarchy upon which additional layers of data protection may be constructed. Continual improvements in disk drive reliability have made possible the design of disk mechanisms with a Mean Time Between Failure Rate (MTBF) in excess of 150,000 hours. However, this large MTBF does not guarantee against failures, but only implies that failures will occur less frequently on average.

Moving up to successively higher layers in the data availability hierarchy becomes increasingly costly, since redundant duplication of hardware is a requirement for fault tolerant systems. However, the middle sections of the hierarchy may offer acceptable levels of data availability for far less cost than the fully "bulletproof" solutions. Many applications can take advantage of this medium level of availability and the lower cost is an attractive feature. These intermediate levels of the availability hierarchy consist of disk storage arrays and disk mirroring products which will be examined in more detail in later sections of this paper.

Increasing disk hardware reliability will have an enormous effect on data availability. However, a high Mean Time Between Failure (MTBF) for a disk drive does not guarantee against disk failure. It represents an average number based on field experience or a theoretical calculation based on component failure rates. High

MTBF disk drives will still fail, but not with the same frequency as they have in the past. When a disk failure occurs, it may bring down the computer system for a long period of time, particularly if the data has been corrupted and a system reload is needed. Ten hours or more of downtime is not an uncommon result when a disk failure occurs. To many customers this amount of system downtime is intolerable.

As the number of individual disk mechanisms in an array increases, the Mean Time Between Failure (MTBF) for the entire disk storage system decreases proportionately. This makes intuitive sense, since the more components that exist in a system, the less reliable the total system becomes. For example, the approximate number for the average MTBF of a disk storage configuration will be the average MTBF for a single disk divided by the number of mechanisms attached to the system. Given these facts, an array consisting of multiple disk mechanisms required a new method to increase data availability as a means to avoid more frequent disk failures and resultant system crashes. This led to the search for a controller design which maintains system and data availability even in the event of a disk failure.

As a result, the parity disk concept was developed, providing the array controller with the intelligence to reconstruct data from a failed disk drive on demand. This results in a data protection shield which renders a disk failure completely transparent to the user. Disk failures in a parity disk array now have no effect on the end user! The array will now provide high data availability, use lower cost disk mechanisms, provide large quantities of storage, and provide performance improvements, all at the same time. Consequently, the disk array is a highly flexible mass storage device by virtue of meeting all these diverse user requirements.

Disk Array Performance

Although the primary function of the array is to provide a large amount of storage at a high level of data availability, in certain modes and applications some performance gains may be realized. Through the use of an intelligent controller managing the operation of several disk drives, it is possible to achieve performance gains from a number of different perspectives. If all spindles are synchronized and the data is striped over all disks, data transfers take place in parallel. Therefore, the array has a potential speed advantage for large data transfers. Instead of one disk transferring data, the array has multiple disks transferring simultaneously. In this mode of operation the disk array appears to the host system as one large disk drive.

In another controller design, the array may appear to the computer system as several unique disk drives all operating independently of each other. This allows for concurrent disk mechanism operation without data striping or parity, useful for very I/O intensive applications, since multiple I/O's can be executed at the same time.

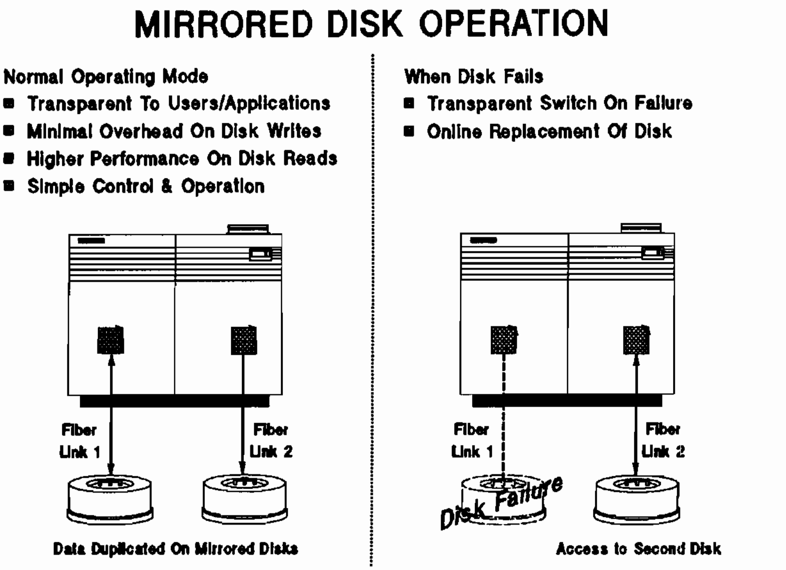
Different types of arrays can also achieve concurrent operation for certain operations by working together in pairs of a much larger group. Arrays of this type can process multiple small transfers simultaneously to speed up performance. In this example, the striping is done on a smaller number of disks in the group for small transfers and large transfers will keep all the drives busy in parallel. This will be explained in more detail in subsequent sections of this paper.

REDUNDANT ARRAYS OF INEXPENSIVE DISKS (RAID)

A common industry buzzword associated with the subject of disk arrays is called **RAID**, an acronym for Redundant Arrays of Inexpensive Disks. Many different configurations of disk arrays are possible, depending on the requirements of the end user and the goals of the manufacturer. Each design has a different functionality built into the controller to accomplish specific goals related to disk performance and data availability. A University of California at Berkely paper entitled "A Case for Redundant Arrays of Inexpensive Disks (RAID)" by Patterson, Gibson, and Katz, summarized five categories of disk arrays. Since only three of these configurations are practical for most on-line transaction processing systems, we will examine them in more detail.

RAID LEVEL ONE - DISK MIRRORING

This classification describes the concept of disk mirroring, where fully redundant disk drives are used to store data on a computer system. Here's a representative sample of a typical disk mirroring storage system:



In the event of a disk failure, the special disk mirroring software will automatically switch all I/O activity for the mirrored pair of disks to the surviving disk in the pair. Repair and resynchronization of the failed disk drive can be done transparently to users and applications. The system knows that a failure has occurred via a console message, but the users and their applications will have the same access to the data as if nothing at all had happened. High data availability is a key user benefit from disk mirroring, since it extends system uptime by saving the system in the event of a disk failure. Mirroring has minimal overhead on disk writes, since two copies of the data and any changes have to be made. Higher performance can be achieved on disk reads however, since I/O can be processed by two disk spindles concurrently. This provides a performance benefit in addition to high data availability.

The disk mirroring solution duplicates the entire disk system, thus protecting the data against power supply, controller, fan, and cabling failures, in addition to a failure in the disk mechanism. The disadvantage of disk mirroring is the cost of duplicating the disk drives, making the effective cost per megabyte twice that of an unmirrored system. Actually, the true cost may actually be more than twice as much due to the cost of the disk mirroring software which manages the entire operation. For data which must be protected at all costs and remain on-line in a high availability system, disk mirroring advantages will far outweigh the higher costs.

Disk mirroring also allows a system to be designed to achieve on-line backup. Since there are two disks storing every bit of data on the system, one disk in the mirrored pair can service on-line transaction processing, while the other can be dedicated for backing up to a secondary storage device. The only delay in the system is approximately five minutes of quiescent time at the beginning of the process to maintain data integrity. After the backup has been completed, the software schedules an on-line resynchronization of the mirrored disks.

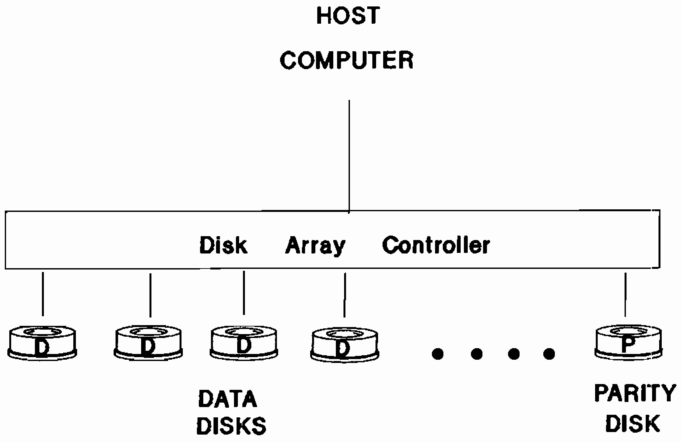
RAID LEVEL THREE - PARITY DISK

This type of disk array uses a separate parity disk to store checksum data. The function of the parity disk is to store the EXCLUSIVE OR of the data kept on the data disks. This allows for a bit-by-bit comparison and subsequent reconstruction of the data in the event of a failure of one of the data disks. The number of data disks in the group is usually chosen to be an even number and will depend on total capacity desired as well as packaging considerations. The disk spindles are synchronized so that at a given point in time all the heads in the group of disks are reading or writing on the same sector location in parallel.

Data is spread or striped across all the data disks in the array on a byte-by-byte basis, and the array appears to the system as a single large disk drive. During normal operation the array transfers data in a parallel fashion at the theoretical rate of a single mechanism multiplied by the number of data drives. Actual transfer rate depends on the host bus adapter bandwidth and system data patterns. For some applications, a performance improvement will be achieved in the resultant higher transfer rate. Every write operation will involve all disks in the array, since new parity needs to be written to the parity disk. Reads involve all the data disks in the array.

Here's an example of a typical dedicated parity disk array configuration. Notice how the parity disk can protect even multiples of data disk drives, even though only four disks are shown.

DISK STORAGE ARRAY
PARITY CONFIGURATION



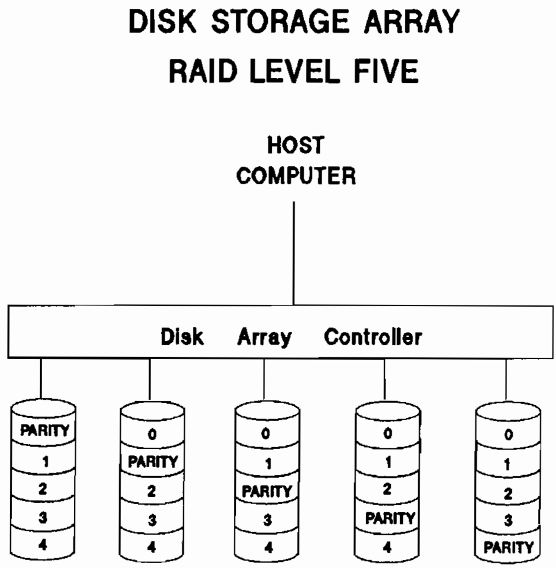
In the event of a disk failure, the array controller reconstructs data that is missing from the failed disk through the use of the parity information. Since all the bytes are buffered, there should be no loss of performance as the controller is specially designed for speed to accomplish the extra tasks involved. The array can operate in this mode with no loss of efficiency until the chance occurrence of another disk failure, but statistically the probability of that happening is very small. The system knows that a disk failure has occurred, and the failed mechanism can be replaced at the next service call or else during a slack period in the system operation. A new drive can be inserted to replace the faulty unit and the array controller will rebuild the data on the new disk. During all this activity, the system is still available for on-line transaction processing applications.

The cost per usable megabyte of storage in this type of disk array increases due to the dedicated parity disk which cannot be used for storage of user data. For example, in a four-way striped parity array, there is 25% overhead in storage costs. This is due to the fact that the product consists of five disk mechanisms, but user data can be stored on only four. The overhead associated with the parity is a small price to pay for the increased system uptime.

Other components of the array can be duplicated to guard against power supply and controller failures. A large amount of flexibility exists for variations in the design of this type of array, depending on user requirements.

RAID LEVEL FIVE - DATA EMBEDDED WITH PARITY ACROSS ALL DISKS

This design is more complex and therefore more costly than the level three disk array. Here's an example of a mass storage array incorporating a level five controller:



The data is still striped across all the disks in the group, but the parity information for each sector is not stored on the same disk. The array controller manages the generation and location of the parity information for each sector stored. For example, the controller parity and data storage sequence might be as follows: for sector zero the parity data is stored on disk one, sector one's parity is stored on disk two, sector two's parity is on disk three, sector three's on disk four, etc.

This level five array may in some applications have decreased write performance, depending on the system, since for a write involving just a few disks in the group for a small block of data, all the disks in the group need to be read, new parity calculated, and then new parity information rewritten. This "READ-MODIFY-WRITE" cycle represents extra overhead for small writes, when compared to a level three design. On small reads involving just a few disks, better read concurrence occurs, since the array may be processing multiple I/O's to different disks in the group.

In the event of a disk failure in a level five array, the missing data is calculated from the parity or checksum information in the same fashion as in the level three array. Data availability is correspondingly increased due to the extension of system uptime, even if a disk failure occurs. As a result, a disk failure does not automatically imply a system crash.

SUMMARY

Disk arrays represent another step upward in the data availability hierarchy, designed to meet diverse user requirements that tend to conflict with the more typical disk storage system. User requirements for increasing quantities of disk space often conflict with the need to achieve higher system availability. The disk array allows the user to obtain larger amounts of storage at higher availability levels.

Data processing managers for large, multi-user systems want to take advantage of the economies of scale inherent in many of the smaller diameter disk drives that have become so prevalent in the personal computer environment. Why must the data center continue to use larger disk drives that take up so much valuable floor space and consume such large amounts of power? The disk storage array allows mid-range and mainframe systems to use larger quantities of small form factor disks linked by intelligent controllers to solve mass storage requirements..

Finally, the disk storage array may offer increased performance in some applications, particularly since the transfer rate is multiplied by the number of drives in the array. Other types of disk arrays may permit concurrent writes and reads, depending on the transfer size. Actual performance of course, depends on the host system performance profile as well as on the applications being processed.

As a result, the future looks appealing for a wide variety of disk array products, each improving on certain aspects of performance or data availability. Many variations of the disk storage array exist today, but the basic conceptual design will easily support many future enhancements designed to increase versatility and flexibility for the end user.

Disk Recording Technology-From DC to Light

Michael Rusnack
Hewlett-Packard Company
Disk Storage Systems Division
11413 Chinden Blvd.
Boise, ID 83704

Introduction

This presentation discusses the evolution of disk drive recording technology. Examples used in this paper mainly refer to Hewlett-Packard products. Even though HP products are highlighted, the same examples can be applied to most disk drive manufacturers.

The goal of this tutorial is to help you to understand the vocabulary associated with the specification of disk drives. From there, using the terms and definitions learned, we will walk through the 20 years of disk drive history-from magnetic recording technology through optical data storage. The goal of this presentation is to first bring to you awareness of the trends in disk recording technology over the years. Second, understanding of the offering of each disk drive model. And finally, to determine what are your specific recording needs and how to select a product to meet those needs, be it magnetic, writable optical or read-only optical.

When reviewing a data sheet, often you will find the phrase "state of the art" or similar claim. What is state of the art? Is it the highest capacity? Fastest transfer? Smallest size? What other specifications should be considered? To begin this 20-year tour through disk recording technology, the vocabulary of the technology must be reviewed. Next the evolution of the technology will be discussed. This will include the growth of magnetic recording in capabilities, and the simultaneous reduction in size of the devices. Finally, an introduction to optical recording technology will be presented.

Terms and Definitions

Disk drive technology has its own vocabulary, unique meanings of words and acronyms like most technologies. As preparation for the more in-depth technical portions of this tutorial, I would like to review some of these terms:

HEAD - The Device used to sense (read) or alter (write) the magnetic signals on the media. Typically, there is one head per surface.

DISK - It is important to understand the proper spelling. Until audio compact discs, the spelling was with a "c." In order to differentiate computer mass storage device from audio devices...

DISK - (continued)   AUDIO/VIDEO - disC
COMPUTER - disK

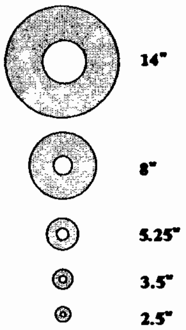
Certainly it is up to the discretion of the manufacturer how they spell the word, however those in the know spell it with a "k."

MEDIA - The material used to record the data. Magnetic disk drives use an aluminum platter coated with magnetic particles. Until recently, the disks were coated with a thin layer of iron oxide. The iron oxide is sprayed on to the disks as it is spun at a high speed. The oxide spreads over the surface, collecting at the outside diameter.

Higher recording densities have been achieved as the result in breakthroughs in THIN FILM media development. The same aluminum disk is used; however, very thin layers of metal are sputtered onto the surfaces. The result is a very shiny, mirror-like surface.

Typically, both sides of the recording surface are used to store data.

Recording media can range in diameters from 14" to 2.5". Industry standard sizes include 14, 8, 5.25, 3.5 and 2.5" diameters. Figure 1 below shows a size comparison of these disks.



Comparison of Media Size

Figure 1

BIT - The term is derived from binary logic, a bit is either ON or OFF, one or zero.

BYTE - Eight bits make up one BYTE. The memory capacity of a device can be specified in either bits or bytes. You can be easily confused if the bits and bytes were confused.

BYTE - (continued) A convention often used when describing the capacity of a device is to abbreviate; two million bytes is 2 MB. Five million bits is expressed as 5 Mb. Note the large B (bytes) and small b (bits).

BPI - Bits per inch

FRIPI - (pronounced fripee) Flux reversals per inch. The last three are measures of data density. Basically, this is a description of how tightly packed is the data on the media. This corresponds to the capacity of the drive.

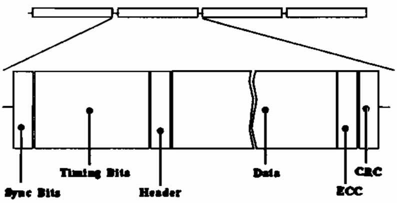
BPI and FRIPI are measures of linear density. The data on the disk is written in concentric circles or tracks. If you were to take the track (circumference) and draw a straight line, this would be the linear density.

The TPI or aerial density refers to how tightly these concentric circles are spaced.

TRACK - A concentric circle of data, typically numbered 0 to N, from the outside (OD) to the inside diameter (ID).

TPI - Tracks per inch

SECTOR - The track is made up of sectors. Each sector contains information necessary for position verification, data and error correction. Figure 2 below shows the typical format of one sector.



Sector Format

Figure 2

The sector has several components. The first bits encountered (after the sector gap) are the SYNC BITS. These awaken timing circuits that use the TIMING BITS for clock synchronization. Next, the HEADER contains address location, including head (or surface), cylinder (or track) and sector number. The DATA field contains the user data. The ECC (Error Correction Code) and CRC (Cyclic Redundancy Code) are used to ensure the accuracy of the data.

ACTUATOR - The actuator is the device that positions the heads on to the media for the purpose of reading and writing the data. There are several types of actuators. Low cost drives, floppies and hard drives alike may use a stepper-motor. This actuator is slow, and its steps are larger compared to linear or rotary actuators. The result is the need for tracks to be wider and spaced further apart.

Extremely high performance actuators use a linear voice coil. The actuator moves in a straight line, front to back. Although high performance movement is achieved, the actuator assembly is large and bulky.

A compromise is the rotary actuator. Like the tone-arm of a record player, the arm rotates at an arc across the surface. Speed and accuracy are present in a much smaller and more compact package.

SERVO - To understand the differences in disk drives, and their evolution, some mention must be made about the servo positioning system. This refers to the system that locates the read/write head over the proper track. The type of positioning system affects three parameters:

Accuracy
Time to move from track to track
Track density

There are several types of servo systems, each have benefits and detriments. Listed below are some types and examples:

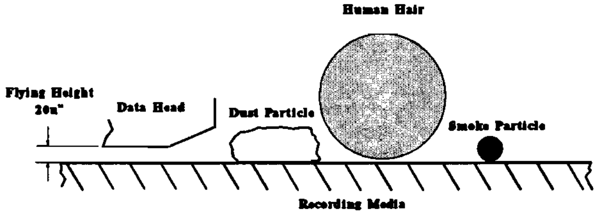
Electro-Mechanical	Stepper motor
Optical	Source/Sensor
Embedded	
Dedicated	Written on SERVO Surface
Sampled	Written with DATA

The stepper motor is used most typically in floppy drives, and the embedded used in hard disk drives.

SERVO - The DEDICATED SERVO SYSTEM uses a "dedicated" surface to contain the position information. The SAMPLED SERVO SYSTEM writes the position information in areas that data is not written, the gap between sectors.

The Magnetic Domain

Early disk drives used removable disk cartridges or packs. These packs allowed users to interchange data much like a floppy system today. Magnetic recording on hard disks is much like tape recording in that a magnetic material is magnetized in specific patterns that represent encoded data. In the case of tape technology, the tape (magnetic media) is passed over the read/write head. The speed of the tape is optimized to minimize wear to either the tape or head. The data on a disk passes by the head at over 1600 inches per second, compared to 30 inches per second for a tape drive. To read and write data at these speeds, the head must not actually contact the disk, however it must be very close. The data head had to "fly" over the data; typical flying heights range from 15 to 25 micro-inches. Just how much is 25 millionths of an inch?. In the figure below, several comparisons are made to everyday items.



Flying Height Comparison

Figure 3

The stable flying of the head is critical. At the equivalent of 90 miles per hour, the head must be fly stable in a clean chamber. To ensure this, early manufacturers required that the media chamber be sealed, purged and filled with nitrogen or even helium. As the technology developed, the sealed/pressurized chamber was replaced with an air filtration and circulation system. A blower and an absolute filter were used to circulate "clean" air through the disk chamber. Often, the disks will spin to purge the chamber of any contaminates before the loading of the heads.

If the head and media were to come in contact, for whatever reason, irreparable damage will occur. This event is called a head crash. Often, the slightest contact will result in the damage of the media surface. The oxide surface is soft and is easily removed by the hard, glass head. This results in the generation of contaminates. The contaminants are generated faster than the filtration system can remove them, thus the crash propagates. The final result is total loss of data; the recording media is literally scraped away.

The early disk recording technology was lead by IBM (International Business Machine). Whenever a "futures" slide was presented, IBM was always one generation ahead of everyone else. Often, the disk technology was referred to by the heads/media interface. Flying head drives like HP's MAC family, HP 7906/20/25 drives used the IBM 3330 technology head design. These heads were relatively large and cumbersome. They literally flew over the disks. Pictured below is a head assembly. At rest, the assembly is bent as seen below. To bend the assembly straight takes 150 grams (or 5.25 ounces). The upward force that keeps the head flying above the media is due to the aerodynamic design of the head assembly.

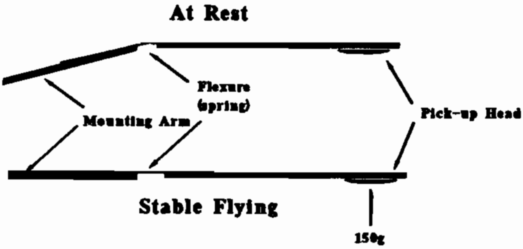


Figure 4

This technology used 14 inch (or larger) platters. With bit and track densities relatively low, the only way to increase storage capacity per spindle was to add platters. A 14 inch disk has about two inches of usable space for data storage. Hewlett-Packard's first offering in this area was the HP 7900; a 14-inch drive using two platters. This yielded 5 MB total storage. This first offering had 100 TPI with 2200 BPI. How does this compare with future offerings? Illustrated in the figures below is the increase of tracks per inch and the increase of data storage per 14 inch surface.

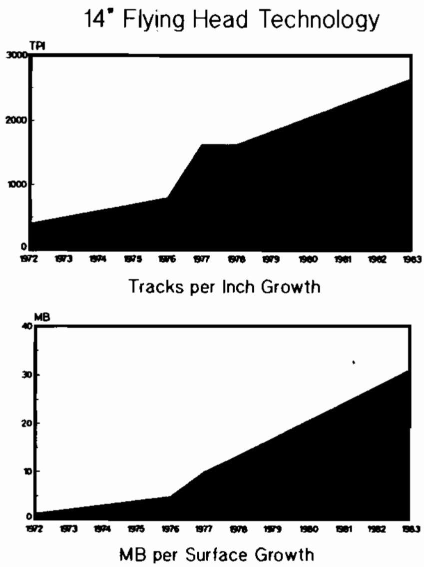


Figure 5

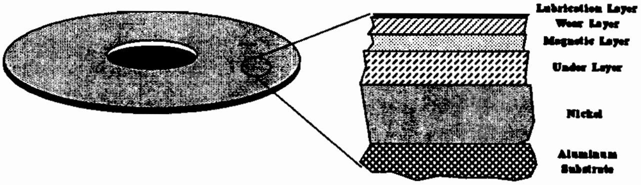
The capacity need of computer users are being increased over the years with the complexity of operating systems, the number of applications available and the increase in CPU power. To meet this growing demand, each disk product has increased the bit and/or track density. to further increase capacity, more platters were added to the drives.

The demand from the disk drive users was not only to increase storage, but to decrease the size of the package. The disk drive was not only examined for it capacity, but its size and power draw were studied.

Again, as with previous entries into the disk drive market, the technology was driven by IBM. This class of drives was called Winchester Technology. The differentiating factor of this technology was that the heads did not retract from the disk's surface at power down. An area where the heads were allowed to reside when the disks were not spinning was designated as the landing zone. This area was typically located at the ID, well away from any customer data. Before the development of the harder, thin-film media, the oxide media was coated with a thin layer of oil that reduced the chance of damaged when the heads came to rest on the disk's surface.

The second distinguishing feature was the heads and media were now sealed in a nearly air tight chamber. With flying heights being reduced and tighter tolerances being maintained, the sealed HDA (Head Disk Assembly) was born. Except for a small vent hole, the heads and media were sealed from outside contamination.

New oxide formulations and smaller head sizes allowed the increase in tracks per inch, thus resulting in even greater storage capacity per data surface. The demand for greater storage per spindle grew, as did the requirements for smaller sized disk drives. To meet these needs, the next step was to thin-film media. These disks start with the same aluminum disk substrate as the oxide, however the magnetic coating consists of several layers of metal deposited on to the surface by sputtering. The first HP product to use this new technology media was the HP 7936/37 family of disk drives. The cross section of a thin film disk is depicted in the graphic below. This is substantially more complex than the oxide media.



Thin Film Media Cross Section

Figure 6

The thin-film media used was 8 inch rather than the previous 14 inch used in earlier products. Though the 8 inch disk had only 1.25 inches of usable area, the capacity per surface was 37.8 MB: a gain of twenty two percent, more storage capacity on a disk that is nearly half the diameter and one third the footprint of previous disk products.

Like its Winchester predecessor, the thin-film disk drive's HDA is sealed. The new technology disks present a new challenge-corrosion. Special filtration systems must be applied to not only remove particulates, but to remove moisture and corrosive chemicals. In parallel with the reduction in the size of the HDA, a corresponding reduction in the overall number of electronic components occurred. Many of the functions performed by hundreds of discrete components are now integrated into a few chips. Through hole electronic components are replaced by much smaller surface mount components to save space and reduce the size of the printed circuit assemblies. Electronics that once were contained on three PC boards are reduced to one board, and are half the size of earlier boards. With the smaller HDAs and highly integrated PCAs, the reliability of these disk drives exceed their predecessors by a factor of ten.

The demand for more storage capacity grows with every computer system. With the increase in competition within the disk drive industry, the focus is on cost reduction. The new technology media is the secret to the increase in MB per drive, however the cost of the new thin-film disks are several times that of the oxide media. The focus is not on capacity per spindle but the cost of that unit. There were two primary methods to reduce cost - use 5.25 inch disks and manufacture in volume. Using smaller disks, more can be produced per deposition machine. Secondly, at higher volumes the cost per disk drops significantly.

Where is the limit? Where will it all end? We have experienced growth in capacity, offering hundreds of MB on a single spindle. Conversely, we have watched the disk size shrink from 14 inch to 8 inches all the way to 2.5 inches. From the table below, the answer may be apparent.

Technology Trend for Magnetic Media			
	TPI	BPI	MB/ Surface
8"	1121	18.8K	22
5.25" - I	1950	20.5K	27
5.25" - II	1667	30.6K	48
3.5"	1850	4.2K	47
2.5"	1456	39.7K	11

Table 1

From the second generation 5.25 inch disk on, the densities appear to have peaked. Indeed, there will be a breakthrough in the head/media technology someday. Vertical recording technology is in the laboratory today, and is said to be the next step in magnetic recording. Until then, or another technology breakthrough occurs, these physical limitations will remain the barrier.

...and then there was light

The days of removable media drives are long past with the advent of the Winchester technology drive. The sealed HDA is not normally used as a removable media device. Applications such as data interchange, archive and security, are not met by the slow, low capacity floppy disk drives. These applications have created

a need and optical storage has come to pass. In addition to having a removable storage medium, the optical drives are lower in cost per MB than their magnetic counterparts. Magnetic storage will not be displaced by optical technology today, however, it has much slower performance.

Data is written (and read) using a LASER as opposed to a magnetic head. Due to the ability of the LASER to be focused to a very small area, the data densities on optical disks are 10 times or more than the equivalent size magnetic disk. Hewlett-Packard's Magneto-Optical disk drive Model 650A, boasts of densities of 15,875 TPI on a 5.25 inch disk. This translates to 325 MB of storage per surface. That is a tenfold increase in track density and over 6 times increase in raw data storage when compared to a magnetic disk.

Features of optical recording technology include:

- -Low cost per MB
- -High data densities
- -Removability of media
- -Longevity of written data
- -Not susceptible to magnetic interference

There are three types of optical storage devices:

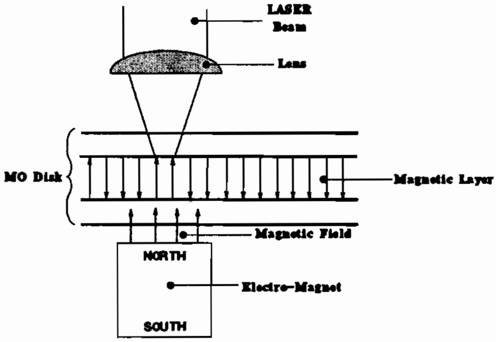
- o Read/Write
- o Write Once/Read Many (WORM)
- o CD-ROM

Read/Write optical devices come in three flavors:

- o Magneto-Optical (MO)
- o Dye Polymer
- o Phase Change

MO media can be erased and rewritten repeatedly, like hard disks. The MO drive uses a high power LASER to assist in the write process, and a low power LASER to read the written data. There is a layer in the MO media that has a very high magnetic resistance.

The magnetic field required to alter the bit direction varies greatly with temperature. The LASER is focused to a small point on the media. This action heats the magnetic layer to its "Curie Point." At this high temperature, the properties of the magnetic layer change, thus allowing the drive's magnetic field to alter the magnetic polarity of the bit. The direction of the magnet's polarity determines whether the bit is a 0 or a 1. This write process is shown in the figure below.



MO Write Process

Figure 7

To read data written on an MO cartridge, the technology takes advantage of a physical law known as the Kerr Effect. This law states that a magnetic field affects the polarity of reflected light. By detecting the light's polarity, the direction of the magnetic field can be determined.

Dye Polymer technology uses a translucent plastic disk with a colored layer which absorbs heat from the drive's LASER. A blister is formed on the area heated by the LASER. Reading the Dye Polymer disk is similar to reading a CD-ROM. The blisters reflect light differently than the flat areas. One drawback of this technology is the media life is limited to less than 10,000 write cycles.

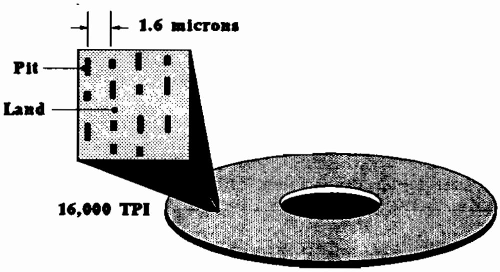
The Phase Change method uses a plastic disk with a special metal layer. Heat generated by the drive's LASER changes the molecular structure of spots on the metal layer from an amorphous state to a crystalline state. To read, differences in the reflected brightness in the crystalline spots are detected. As in the Dye Polymer method, the Phase Change disk has a finite limit to the number of write cycles.

WORM or Write Once Read Many is another type of optical disk drive. WORM uses a disk with a special metal layer. Heat generated by the drive's LASER alters the media surface. Like the process noted above, the WORM media can be written only once. WORM is used for archive for material that will not change, such as insurance data, title history or printed documents. Since the WORM data cannot be altered, it is secure and considered a permanent record.

Computer based CD-ROM technology is based on the home stereo mechanism. This read only device can store up to 550 MB of computer data. That is the equivalent of 200,000 printed pages of text, 5000 high quality color images or one hour of CD quality audio.

Data on the CD-ROM is organized in equal-length sectors that spiral from the inside to the outside diameter of the disk. This differs from the magnetic disk, as well as its optical counterparts, in that both magnetic and optical are written in concentric circles. The format of the 'digital' CD is the same as the conventional CD that stores music. The result is that audio is integrated with the digital data.

Shown below is how information is recorded onto a compact disk.



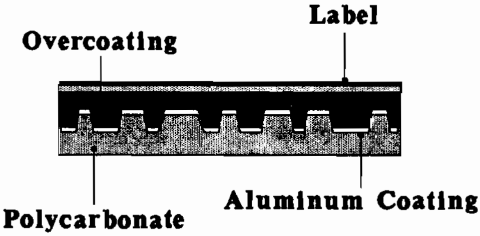
CD-ROM Data Pattern

Figure 8

The ones and zeros are recorded as PITS. The flat areas between the pits are called LANDS. The continuous spiral spaces the tracks 1.6 microns apart. This results in a track density of 16,000 TPI, 10 times more dense than magnetic recording.

The process to produce CD-ROM media is relatively low cost and simple. The data is mastered from any of a number of sources, tape, floppy, even magnetic disk. When the preparation of the data is complete, the formatted CD-ROM image is cut into a glass disk using a high power LASER. The glass master is then taken to an electroplating process where a metal stamper is made. Using the metal stamper, an injection molded disk can be produced every few seconds. The base material is then placed in a metalization process where it is coated with a reflective aluminum layer. The aluminum layer is covered with a protective layer of clear polycarbonate.

Pictured in the graphic below is a cross-section of a compact disk.



CD-ROM Cross Section

Figure 9

Encyclopedia Britannica recently introduced "Compton's MultiMedia Encyclopedia" that includes spoken segments of famous speeches, classical music and a 20-minute glossary of terms. The CD-ROM set-encyclopedia, with 31,000 articles, 15,000 photographs, charts and diagrams is only one of the many applications of this young technology.

For many years, IBM has led the charge in computer technology. Today, there are many companies with R&D efforts in this field. Although "Big Blue" manages to be on the forefront of technology most of the time, the rest of the world is right there too.

This presentation has provided you with a brief overview of the evolution of disk recording technology. To cover every aspect of recording technology would take hours. It is my intention that this presentation has provided you with new knowledge and understanding of this technology. With this new awareness, I hope that you will view your disk drives as more than just data storage.

Image Processing: How Should it be Utilized?

Authors: Brent M. Ehrman, Manager
Crowe, Chizek and Company
2100 Market Tower
Indianapolis, IN 46204-2976

Paul S. Thomas, Associate
Crowe, Chizek and Company
2100 Market Tower
Indianapolis, IN 46204-2976

We have all seen pictures in school text books of rough drawings of horses and cattle on a cave wall. From these simple drawings, scientists have gathered volumes of information about the artist and the environment in which this artist lived. Since the early caveman days, the old adage "A picture is worth a million words" has held true.

Within the last five years, the power of a picture (or an Image) has gained renewed interest within the computer industry. Traditionally, computers have only been able to store and manipulate numbers and characters electronically. Obviously this has allowed businesses to perform certain tasks much more quickly and efficiently than using older manual methods. However, it is still common to hear complaints from users that manual records must still be maintained because information from forms, hand written notes and letters, pictures, etc... are needed to document the entire business transaction.

Prior to the major hardware improvements of the past few years, computers were not able to process images due to the significant amount of system resources consumed by each image (e.g., one 8.5" by 11" document may require from 25K to 40K even after complex data-compression techniques have been applied). However, today's computers have the storage capacity, communication speeds, and processing power to process images of entire documents just as they have processed numbers and characters for years. Just as traditional number and character based computer applications have revolutionized the way in which businesses process information, Image Processing applications promise to have the same type of overwhelming affect in the future.

IMAGE PROCESSING PRODUCTS

Practically every major computer vendor has introduced an Image Processing product within the last five years. For many vendors, these new imaging products have only been released within the last year. The following table lists just a few of the many vendors who now provide an Image Processing solution.

Company Name	Imaging Software	Hardware
AGA (Advanced Graphic Applications Inc.)	DISCUS (Data, Image, Sound, Communications, Unified Storage)	80286 MS-DOS/OS2
Bell & Howell	Image Search Plus	80286 MS-DOS
Chorus Data Systems	Docutrieve DT-2000	80286/80386 UNIX
Digital Equipment Corporation	DECimage	VAX VMS
Eastman Kodak	KIMS (Kodak Information Management System)	VAX VMS
FileNet	Document Image Processor Series	Motorola 68020 UNIX/MS-DOS/OS2
Hewlett Packard	AIMS (Advanced Image Management System)	HP1000/HP9000 HPUX
IBM	ImagePlus	System/36 AS/400 MVS/ESA systems
Imnet	Imageview Image Storage and Retrieval System	80286/80386 MS-DOS/OS2
LaserData	LaserView	80286/80386 MS-DOS
Litton Industries/ Integrated Automation	Docuvision	UNIX/VMS/ MS-DOS/OS2
Micro Dynamics Ltd.	MARS	Macintosh
Summit Software	Summit Workflow	UNIX/MS-DOS
TAB Products	Laser-Optic Filing System	80286/80386 MS-DOS
3M	Docutron 2000 Optical Disk System	80286/80386 MS-DOS
Wang Laboratories Inc.	WIIS (Wang Integrated Image System)	VS series VS/OS

This is not intended to be a complete list; however, you can see that there is no shortage of vendors who are interested in the Image Processing market. Most vendors see this market as one of their few growing target markets. Therefore, a number of vendors are putting forth major efforts to establish themselves within this market. The huge show put on by vendors at the recent Association for Information and Image Management's (AIIM) conference at the McCormick Place Convention Center in Chicago demonstrated the potential they visualize within the Image Processing market.

All of this interest and competition among the computer vendors is bound to benefit the end users. The influx of new software designed to help businesses manage their electronic documents should grow rapidly within the next year or two. Now that the hardware can handle the imaging requirements, there is a need for new types of software which will allow users to take full advantage of the images. There are a number of different types of Image Processing systems which provide varying levels of integration into a businesses day to day operations. Electronic Filing Cabinets and Integrated Image Processing Systems are two specific types which are important to discuss.

ELECTRONIC FILING CABINETS

Most people probably think of an electronic filing cabinet when they think of an Image Processing system. Instead of walking over to the paper file cabinets and searching through an alphabetically or numerically sequenced file, an electronic filing cabinet allows the user to simply type in a key identifier and an image of the document is displayed on the screen. This is extremely beneficial to many companies, especially large companies which are spread out over large geographic areas. Retrieval time of information can be reduced from hours or days down to seconds.

In addition to improved retrieval time, electronic filing cabinets also significantly reduce the number of lost documents. An electronic filing cabinet has the ability to maintain a number of keys by which the document may be accessed. For example, in a manual filing cabinet, invoices are typically filed in alphabetic order by customer name and then numerically by invoice number. Therefore, to find a copy of the invoice, the customer's file must be pulled and then the invoice should be in that file in the proper numerical order. However, should an invoice be accidentally misplaced into the wrong customer's file, it is practically impossible to find the invoice. An electronic filing cabinet allows you to attach multiple keys to the invoice. Therefore, should the wrong customer number be entered into the system, you would still be able to access the invoice directly by invoice number, or customer po-number, or salesman, or whatever keys are appropriate.

Of course, electronic copies of documents require significantly less physical storage space than the actual paper copies. Also, electronic documents can be managed by fewer people than paper files. Often times, the savings associated with freeing up expensive office or warehouse space and reducing the number of document management personnel is enough to cost justify the purchase of an Image Processing system.

INTEGRATED IMAGE PROCESSING SYSTEMS

There seems to be a misconception among many people that electronic filing cabinets are the only type of Image Processing systems. Although the benefits associated with an electronic filing cabinet are worthwhile, and by themselves have served as the justification for a number of existing Image Processing systems, these benefits are just the tip of the iceberg.

The real power of an Image Processing system is the effect it can have on the method or sequence in which documents are processed throughout an organization. An Image Processing system should be used to speed-up the time it takes a document to flow through the organization. It should be tightly integrated to a businesses' traditional numeric and character based applications. Most electronic filing cabinets are stand alone applications which do not have any type of electronic interface to the company's traditional business software. By speeding up a document's flow through an organization, an integrated Image Processing system has the potential to significantly benefit the profit centers within that organization.

For example, data entry is one area where an integrated Image Processing solution can speed-up the process and reduce the number of data entry errors. There are a number of systems available which assist data entry personnel by displaying windows which contain a copy of the original document. As the data entry person tabs from field to field on the data entry screen, the image window changes to highlight the data on the original form which applies to each specific field.

For another example, it is common to find a document which must go through a specific review process. In a paper oriented office this review process is typically a single threaded process where no two people can review the document at the same time. Each time the paper folder is moved from desk to desk, there is a higher probability that it will get lost in the shuffle. If someone would need to retrieve information from this folder while it is being routed around the office, he would have to walk from desk to desk and search through each persons in-box, "to-do" stack, open folders on the desk, etc... In most organizations, there is no way to quickly determine the status or location of a document like this.

An integrated Image Processing system would allow any number of people to access the document at one time (assuming they all had security access to the document). There are a number of applications available which help manage the workflow of documents throughout an organization. For instance, they provide the capability to set-up specific sign-off points which must be performed before the document moves on to be reviewed by the next person(s). At any time, a person would be able to access this document and determine its status, determine who has yet to review the document, and determine who is holding up the process. This scenario would fit a variety of activities ranging from processing a loan application, authorizing an employee's reimbursement request, processing a student's enrollment application, or reviewing an insurance claim.

IMPLEMENTATION CHALLENGES

A trap that many businesses fell into when they implemented their traditional number and character based computer systems was to develop/customize the software to process the information in the exact same fashion as they did with their manual system. People's resistance to change and that common attitude of "Well, that is the way we have always done it" seemed to prevail. In some businesses this "Let's not change what we know works" attitude was the proper implementation approach; however, there are many businesses where this type of attitude severely limited the potential benefits which could have been achieved.

Implementing Image Processing systems provides the same types of challenges. In order to realize the biggest benefits from an Image Processing system, an in-depth analysis needs to be performed to determine what manual procedures can be eliminated or reduced as a result of having access to electronic images of documents. The path that a document takes through the organization may be entirely different when utilizing an imaging system compared to the path(s) it takes in a manual system.

For example, in a manual system, it is common for the original document to be a multi-part form. Once the original information has been written or typed onto the form, the carbon copies are separated and sent on their separate ways. One copy may be filed instantly for safe keeping, another copy may be sent to accounting to check a credit history or to set-up accounts, and a couple of copies may be sent to "processing" where they may go through several steps and then are separated and sent to different departments. The "processing" department may have to wait for accounting to sign-off on the document before they can actually begin processing. Notification from the accounting department may take hours to get to the "processing" department.

With an Image Processing system, a copy of the original would not need to be placed in a manual paper file cabinet. An electronic mail message could be sent to accounting and to "processing" to notify them that there is a new document which needs to be processed. At that point, both accounting and "processing" could access a copy of the image at the same time. Once accounting signs off on the document, the "processing" department would automatically be notified to begin. This is a simple example of how an Image Processing system could effect the document flow within an organization. With a little imagination, it is not hard to relate this example to almost any organization.

CASE STUDIES

The best way to understand the challenges and benefits associated with implementing Image processing systems is to analyze case studies of those who have already implemented Image processing systems. The following is a brief summary of two case studies presented in Datapro's "Reports on Document Imaging Systems - Users Perspectives[1]". Notice that these two companies are of significantly different sizes with significantly different requirements.

USAA (United States Automobile Association)

Company Overview: USAA started as a small automobile insurance agency for military officers. It has grown to become the United States largest mail-order firm, which offers a number of insurance products to approximately 2 million members. USAA has achieved this growth with no sales force in the field. All sales are made by mail or telephone. They receive approximately 86,000 letters per day. This is small in comparison to the 300,000 outbound letters, policies, and bills which are mailed out each day. Copies must be kept of approximately 150,000 of these outbound documents.

Pre-implementation: The amount of manual effort required to manage this large number of documents was overwhelming. USAA used 161 full time employees to maintain the manual paper files. 2,000 service representatives entered the appropriate information into USAA's mainframe before the documents were actually filed.

One years worth of documents were kept "active" in a 39,000 square feet warehouse. Approximately 1 out of 10 of these documents would be retrieved from the "active" warehouse. A standard request for retrieval would take several hours. High priority requests would require approximately 6 minutes to retrieve. Obviously these document retrieval times made it difficult for a customer service representative to be responsive to a customer's needs. In a best case scenario, a customer would have to be put on hold for five minutes before the customer service representative would have the documents needed to answer questions. In most cases, and answer could not be provided until the next day.

Post-implementation: "Everything changed in the correspondence handling process except perhaps the initial few steps.... The system has radically changed how service representatives process correspondence. Formerly, customer service representatives would riffle through a stack of papers on their desks and pick one at random to start working on." USAA's imaging system has the ability to prioritize documents based upon who the representative is and what the supervisor's priorities are. Therefore, service representatives simply process documents in priority order. "Service representatives call up the necessary documents on their image retrieval workstations and perform their functions completely without handling paper."

Retrieval time for documents being processed for the first time which are stored on magnetic media is 2 to 5 seconds. It only takes 15 to 30 seconds to retrieve those documents which are less than 1 year old and are stored in an optical jukebox (remember, these were the "active" documents which took from 5 minutes up to several hours in the paper file warehouse). Documents older than one year are kept on optical disk drives which must be retrieved and mounted manually. This typically requires only a few minutes.

Cost-Benefit Analysis: USAA expects to spend a total of $12 to $14 million for a 1,300 workstation imaging system. They estimate that they will save $5 million per year on file cabinet space, office space, and supplies. They have also reassigned document management personnel to other tasks. They expect to have a 2 to 10 percent improvement in productivity of the clerks and service representatives. This represents $2 to $10 million savings per year.

"In terms of serving customers, the key benefit is that a service representative can knowledgeably discuss cases almost immediately. Formerly, when a customer called about a letter written the week before, that letter might have been on any of 2,000 desks or among a stack of 50,000 letters on a shelf. The service representative usually had to ask the caller to recount the details. Now, once a customer is identified.... the document is usually available on the screen within 5 seconds."

Implementation Problems: USAA was forced to document the sorting and prioritizing process which they went through in their manual system in order to develop an integrated imaging system which could do this sorting and prioritizing for them. What they found, was that this process had never been documented before and was being done subjectively. It was originally thought that there were 200 to 300 different categories for all of their documents. When in actuality, they ended up with 650 different categories.

USAA decided not to convert their seven-year backlog file. They estimated that it would take 100 people 10 years to convert all of these documents. Therefore, these backlog documents are only scanned into the system whenever they are needed.

PNC Financial Corp., Education Loan Center

Company Overview: PNC Financial Corp. is one of the nation's largest bank holding companies. The Education Loan Center, located at the Pittsburgh National Bank, is responsible for student loan processing for 15 of PNC Financial Corp.'s member banks and their affiliates. They process 50,000 student loans per year and service approximately 120,000 outstanding loans. During the peak season (May to September) they process an average of 5,000 documents a day. The average life of a student loan is 15 years. Problem customer's files may have to be pulled numerous times throughout the life of the loan.

Pre-implementation: Prior to implementing an imaging system, PNC had 5.5 full-time clerks with annual salaries totaling $150,000 simply moving paper around the office. Their primary objective for implementing an imaging system was to increase productivity by eliminating wasted time searching for old documents.

Operators worked through a stack of paper forms which often times required them to leave their desk to search paper files by hand. The average retrieval time was 10 minutes.

Post-implementation: With the imaging system in place, 2.5 full-time clerks are devoted to paper handling instead of 5.5 as before. These 2.5 clerks are able to scan up to 5,000 documents per day which allows them to handle the peak season workflow. During slack times, 6 clerks continue to convert outstanding loan documents at approximately 10,000 per day.

Now, PNC's imaging system provides a prioritized work queue for each operator. They can call up electronic images of the documents they need to see regardless of the type of transactions they are working with. The average document retrieval time is 30 seconds.

Cost-Benefit Analysis: PNC's total cost for their imaging system was approximately $750,000. They plan to save $80,000 by simply reducing the number of processing clerks from 5.5 to 2.5. Without the imaging system, PNC would have expected to hire four new customer service representatives. With the imaging system, they will be able to handle the workload with their current staff; therefore they recognize a $120,000 per year savings in this area. The paper file system occupied 3,500 square feet, but the new imaging system only occupies 25 square feet. Therefore, they estimate floor space rental savings to be $65,000 per year. In total PNC expects to save approximately $265,000 per year.

Implementation Problems: PNC is using a LAN Image Processing solution. They are planning for 15 to 20 percent growth per year. They are afraid that the LAN will not be able to handle the increased network traffic associated with this growth. Additional network servers may be required.

They currently have only one optical disk jukebox which leaves the loan center totally vulnerable to equipment failure. They are attempting to persuade other departments to purchase imaging systems with optical jukeboxes which they could use as an emergency backup system. They may need to purchase an additional jukebox and add it to the network.

SUMMARY

These two case studies are good examples for demonstrating the type of benefits which can be achieved through an Image Processing system. As was shown in these case studies, Image Processing systems are not inexpensive solutions. A significant amount of analysis must be performed to guarantee that the system is a justified purchase. Although these are expensive systems, companies which are burdened with paper intensive processes can cost justify them a large percent of the time.

Besides the issues of cost justification, Image Processing systems are becoming necessary to help businesses compete in the market place. "Speed to market" is becoming more and more critical. The company which can perform the task the quickest is going to be the company which gets the business. It has been estimated that 95 percent of the information needed to manage a business is still maintained in paper files. Image Processing systems will help to put this information at peoples' fingertips to increase productivity and thereby improve the company's competitive advantage.

Bibliograhy

[1] Datapro, "Section 4 - User Perspectives", Datapro Reports on Document Imaging Systems, 1990, p.51-70.

Protecting Your Data from System Failures

by

Gilles Schipper
G. Schipper & Associates Inc.
P.O. Box 310
Thornhill, Ontario L3T 4A2
Canada
Tel: (416) 889-3000
Fax: (416) 889-3001

1. Introduction

What do you do when you have a system halt, hang, or failure? If you're data processing installation is like most, once any obvious hardware or software problems have been corrected, the system is re-booted and life goes on as normal.

If you're lucky, any resulting data integrity problems will manifest themselves soon after the failure. You will then have to undergo some kind of data recovery process which may or may not be time-consuming and painful. If you're unlucky, days, months, or even years may pass before you pay for your past sins. Usually, data recovery at that point is more difficult.

Often the major stumbling block to effective data recovery is the difficulty in ascertaining whether or not a data integrity problem exists.

This paper deals with application design techniques which can be - and, if possible, should be - implemented, to enable quick detection of data integrity problems following system interruption, and help minimize the effort necessary to recover from resulting damage.

Specific examples included in this paper assume the major data management facility utilized is TurboIMAGE, although the principals would almost certainly apply no matter what data management tool is used.

The terms TurboIMAGE and IMAGE are used interchangeably.

Where differences exist between MPE/VE and MPE/XL operating systems, they will be identified.

2. Where's the Beef?

Many applications today are not designed with the degree of robustness possible with little additional effort during the design stage. External factors which lead to application interruptions are often not considered in the application design phase, often because they are not deemed to be the responsibility of the design team. Instead, hardware failures, or interruptions due to other external factors (such as exhaustion of available disk space) are arbitrarily assigned to the "System Management" or "Operations" group.

Naturally, in the event of a system interruption, the System Management group is usually called upon to deal with the problem. Once dealt with, the objective is to have the system operational as quickly as possible. More often than not, with many users anxiously awaiting system resumption, there is usually added pressure on the system management group to permit system access as soon as possible - after whatever caused the problem in the first place is dealt with. Of course this means that any data integrity problems which could have been caused by the prior failure are ignored - to be dealt with at some time in the future, time permitting - or, the hope is, not at all. You know what they say - out of sight, out of mind.

Unfortunately, this attitude is short-sighted and dangerous. It could result in more serious problems, exacerbated by continued inattention to possible data corruption.

3. The Problem.

Imagine 50 users all running the same data entry program designed to place orders keyed in by the terminal users into a TurboIMAGE data base. Each individual order comprises a single order header, and one or more line item details. As is usually the case with on-line order entry applications, inventory information in the data base is updated as each line item is entered. Not an atypical OLTP (On-Line Transaction Processing) application for an HP3000.

A system interruption of any kind can cause a variety of problems. The likelihood of these problems occurring increase significantly as the number of users modifying the data base increase.

Physical Integrity

Broken chains are always a distinct possibility - particularly, on MPE/VE systems, with Intrinsic Level Recovery (ILR) not enabled. Broken chains occur when the pointers embedded in a master/detail chain, occurring both in master and detail data entries, are out of synchronization. Even with ILR enabled, system failures caused by damage to the physical disk medium will not necessarily guarantee physical integrity. On MPE/XL systems, where ILR is unnecessary to prevent broken chains, the same considerations apply regarding the disk medium.

Logical Integrity

Often, even more important for the health of your data is logical integrity, if for no other reason than physical damage is usually easier to spot. Logical data problems are typically more insidious and often rear their most telling damage later, when solutions are more difficult.

Logical data integrity is compromised whenever a "logical" transaction only partially completes. In the order entry example cited above a logically complete transaction comprises all the appropriate IMAGE DBPUT's, DBDELETE's, and DBUPDATE's associated with adding 1 order header, all the corresponding line item details, together with the necessary DBUPDATE's of the relevant Inventory masters. Since this single logical transaction can span several minutes in real time and up to 50 of them can be taking place simultaneously, chances for logical data integrity corruption are virtually 100% if system interruption occurs during normal working hours.

If nothing is done to find and correct the partial data, problems will occur later on during the invoicing stage or inventory reconciliation.

4. Some Solutions

Following are different ways to address the problems outlined above:

The Do-Nothing Approach

As indicated above, many computing environments simply correct the problem which caused the interruption and immediately enable system access without even a cursory glance at the potential data damage caused.

Typically, much time spent by the dp professionals in those shops consists of fighting the fires that surely must arise - or they happen to be very lucky, and never experience system interruptions.

Needless to say, this methodology is not the recommended one.

The Do-Everything Approach

There are variations in this approach for those who consider that no effort should be spared to protect the integrity of one's data.

The first variant subscribes to the principal that it's always better to be safe than sorry. To that end, transaction logging is enabled for the appropriate data bases, and, in the event of a system interruption, a roll-forward recovery is ALWAYS performed. A roll-forward recovery requires RESTORE'ing the data base(s) from the previous full backup and running DBRECOV.PUB.SYS against the data base(s) using the log file(s) corresponding to the data base(s).

This method can offer excellent system performance as a side-benefit, because if you are always going to roll-forward recover in the event of system interruption, you may as well enable your data base(s) for Autodefer, which provides optimum system throughput - particularly for update-intensive applications. Autodefer enabling is accomplished by running DBUTIL.PUB.SYS and issuing the following command: ENABLE <u>database</u> FOR AUTODEFER, where <u>database</u> is the name of the IMAGE data base.

An additional benefit derives from the ability to exclude modified data bases from partial (incremental or differential) backups. The files representing IMAGE data bases could normally consume as much as 90% of the contents of your partial backups. The means to do this properly are non-trivial but entirely possible with the use of contributed or purchased tools, and careful implementation and maintenance of the transaction logging cycle.

For this method to insure both physical AND logical data integrity, it is **absolutely** necessary that the application be designed to incorporate logical transaction processing (LTP) - with the PROPER use of calls to DBBEGIN and DBEND. Optionally, DBMEMO calls can be included to provide additional data, which could be potentially used by a user-written recovery program (say, to support the recovery of non-IMAGE data structures). Without LTP, only physical integrity can be assured with roll-forward recovery.

Though not a bad policy, roll-forward recovery is simply not practical for many shops simply due to the length of time required to do it. Also, many applications are not properly designed to include LTP, and retrofitting would be necessary - unless one were satisfied with physical data integrity only. Where the applications are third-party software packages, it may simply not be possible to incorporate LTP.

Those who require very fast availability of their applications upon system resumption, in addition to data integrity protection, can consider implementing Transaction Logging together with roll-back recovery. This requires the enabling of Intrinsic Level Recovery, and the disabling of Autodefer. This combination of IMAGE settings is the least desirable from a system performance point of view, but offers excellent data integrity protection, and fast system availability after system interruption.

In the case where additional protection is required due to the significance of MPE and/or KSAM file structures, third-party tools, such as Intact from Carolian systems should be considered. Intact creates transaction logs for non-IMAGE data structures and includes recovery procedures.

A Possible Happy Medium

For those who are quite happy with the do-nothing approach, you can say "ignorance is bliss".

For those installations who have control of their own source code and cannot afford either version of the do-everything approach because: (1) the time required for roll-forward recovery is unacceptable, or (2) the performance drawbacks of roll-back recovery are unacceptable, there's an alternative which is not difficult to implement or retrofit.

It involves programming your applications to keep track of logical "transactions in progress", by placing a record in the data base at the beginning of the logical transaction, and deleting the record from the data base upon termination of the logical transaction.

Now, when any kind of system interruption occurs, you can very quickly determine the exact nature of the physical and logical integrity of your data base by simply examining the "transaction-in-progress" record markers which still exist in the data base. These record markers contain enough information (as described shortly) to allow very fast corrective action to be taken and system resumption to proceed with a minimum amount of delay, while preserving data integrity.

The record markers would be contained in a master data set. The data entry key would be a combination of a 2-character (or longer, if necessary) transaction type (e.g. "51" representing an "order add" transaction) and the transaction key - whatever identifier uniquely identifies this transaction - such as order number, for example. Additional information in this record marker could include user-id and logical device number (from the "WHO" Intrinsic), date and time of transaction beginning, etc.

This information would not necessarily require the creation of a new data set, or data base. You will almost certainly find that your application data already provides a data set (or data base) for "system" related information, such as security, order number seed, invoice number seed, etc. The record markers described above, could easily be incorporated within these already existing data structures.

Those who complain of the potential increase in disk I/O associated with this technique can rest assured that the additional I/O is minimal. Remember, each individual record marker is very small in size (between 10 and 30 characters), and, since they are maintained in a master data set whose capacity is probably no more than 200 (assuming 100 active users and lots of empty space to minimize secondaries), most of this data will likely be kept in memory, thus even further minimizing I/O overhead.

The real benefit associated with this approach is not so much what it **does** for your data integrity. Rather, it lets you gain control of the situation following system interruption very quickly and simply, at the exact time when quick and simple solutions are required.

Additional advantages accrue as well.

Logical transaction processing becomes an automatic and significant byproduct associated with the implementation of this logical transaction record marking technique. If it was difficult to implement IMAGE transaction logging PROPERLY before, it now becomes almost automatic. All that is required is to include the necessary calls to DBBEGIN and DBEND to book-end the corresponding record marker DBPUT and DBDELETE calls.

Your applications become easier to trouble-shoot and debug, because the record markers in your data base often point to the exact nature of the programming problem.

If, instead of deleting the record markers upon transaction completion, you output them to some permanent (or quasi-permanent) data structure (such as a circular IPC file), your programs now have built-in performance data which can easily be processed to produce application performance and productivity reports, with varying levels of detail. Depending upon the data captured in the record markers, you can produce information such as no. of transactions processed per time unit, per user, etc., showing peaks, means, and medians, standard deviations, etc. All without purchasing an expensive application measurement tool.

Conclusions

A little foresight during the application development cycle can go a long way in both diagnosing data integrity problems after system interruption, and in correcting these problems immediately upon system resumption.

The techniques described above are not necessarily a complete solution for all.

For those who absolutely must have quick system availability after a system failure, roll-back recovery and/or additional tools such as INTACT, from Carolian, or various IMAGE mirroring tools, such as Silhouette from HP, or Backchat from Proactive Systems can also be utilized.

No matter what the application though, maintaining close track of "transactions-in-progress" can never hurt.

****

The Technology of Data Compression and its Benefits to HP3000 Users

David B. Wiseman
Proactive Systems
4 Main Street, Suite 101
Los Altos, CA 94022
(415) 949-9100

Introduction

Data compression refers to the recoding of a set of information so that it can be represented using fewer total bytes. This is to be distinguished from other methods of data size reduction such as purging unneeded files, clearing files of obsolete records, summarizing data (for example, consolidating weekly accounting data into monthly data) or abbreviating data (for example, identifying employees by last name and first initial rather than full name). These latter techniques are not true data compression because some data is lost; the original information set cannot be recreated from the so called "compressed" file. For ease of reference, in the remainder of this paper, we will refer to a "set of information" as a "file". Furthermore, since compressed files also become encrypted, we will use the terms "compression" and "encryption" interchangeably.

The Benefits and Trade-Offs of Data Compression

Why use data compression? There are several obvious benefits. Firstly, hardware costs can be cut back because of the reduced requirement for disc drive units. Secondly, given a fixed amount of disc space, more data can be kept on-line. Thirdly, the speed of data transfer can be increased while reducing costs when copying files to disc or tape, sending data over communications equipment, and shipping data recorded on disc or tape media. Fourthly the amount of media (e.g. tape) to archive the data off-line can be reduced. Lastly, as an artifact of the compression process, files become more or less encrypted, depending upon the algorithm used. Therefore data has greater protection from unauthorized access when stored or transferred in compressed form.

The price or trade-off for the above benefits is basically time. The more effective compression algorithms generally need to do something with every word or byte of a file, so there is significant cpu overhead when compressing a file. Conversely, to access data in a compressed file, it obviously must be decompressed beforehand. Most compression algorithms are a "start to finish" operation; that is, the entire file must be processed as a unit. Therefore the entire file must be decompressed in order to access any portion of it. In the past this normally made it impractical to use compression with production files – but see later how this problem has been overcome.

As an additional drawback to compressing information, it might be argued that compressed files are more susceptible to corruption. Particularly with start-to-finish algorithms, decryption requires a precise sequence of operations, which is exactly the reverse of the encryption sequence. If this sequence is disrupted by a few corrupted bits on the storage media, it is quite possible to lose the remainder of the file. However the reliability of current storage hardware renders this risk rather small. Indeed, if your storage unit has deteriorated to the point where it is losing bits and bytes, you are likely to be experiencing far greater problems than a compressed file that can't be retrieved.

Some typical compression figures for HP3000 data types are:

	File Size Reduction
IMAGE Data Bases	60 to 90%
KSAM Files	70 to 90%
Text Files, Source Code	60 to 75%
MPE Data Files	50 to 70%
Object Programs, SLs	0 to 80%

For example, we compressed one of our operational IMAGE data bases using a fairly simple compression algorithm from 123,168 Sectors (31.5 Megabytes) to 36,112 Sectors (9.2 Megabytes) – a compression ratio of over 3 to 1. This data base contains accounting information and name and address records. The sets in the data base varied between 80% and 100% full which is higher than for many operational data bases.

The Compression Process

There are basically two activities associated with all methods of data compression: removing "noise" and removing redundancy.

NOISE: Most files contain varying amounts of "garbage", that is, areas which contain no useful information. This may be due to uninitialized or unused fields, inactive or deleted records and logically dead areas due to operating system characteristics such as fixed length records with inefficient blocking factors. A classic example of this on HP3000 systems are the "spare" entries in an IMAGE detail data set which are only there to avoid the risk of overflowing the set capacity. These areas can be removed if the decompression program can determine how much garbage to restore, and where to restore it when recreating the original file. Here too, a performance trade-off can be encountered in that not all types of randomness may be obvious to the compression program. For example, is it a binary number or is it just a random string of bits? The program therefore must be made sophisticated enough to distinguish data from non-data, and the result of this is increased code size.

REDUNDANCY: With most file types, some recurring patterns of bytes or words can be found. These can be replaced in the compressed file with some sort of symbol which indicates to the decompression program, the particular pattern to restore at that location.

The simplest and most common pattern, regardless of file type, is a string of repeating single characters or binary words. In text files, these are most often the strings of blanks which occur between words, statements and paragraphs. Data files usually contain a significant portion of records with fields that have been initialized but are as yet unused. These will typically contain strings of ASCII or binary zeros in numeric data fields and strings of blanks for text data fields. Moreover, inefficiently blocked files and datasets are padded to the end of the physical block with the last character of the last logical record. This type of redundancy is easily removed, and forms a major component of most compression algorithms.

Apart from repeating strings, other forms of redundancy tend to be more file type specific. With text files, the ASCII text characters obviously occur with great frequency. Furthermore, these characters generally form textual words and word sequences which recur within the file. COBOL source code for example is partially composed with a known set of reserved words which occur with great frequency within each program. If your shop has coding standards which are controlled to the degree that standardized variable names are used, the number of patterns within each file will increase further still. Data files may also contain a substantial number of patterns. Large account numbers may only vary by a few characters. Inventory files may have many items with similar descriptions and identical prices. Although binary files tend to have far fewer patterns, even here they will contain some sequences of identical or repeating machine instructions.

When a file is compressed, it becomes split into two components, the encrypted data itself, and the information needed to interpret the same in order to restore the original file. This latter information will be referred to as an "encryption map". There is a roughly inverse relationship between these components in that achieving greater amounts of compression generally requires a more complex map by which to decode the compressed file. In theory, a file of any size could be compressed to a single element, but the encryption map would possibly approach or exceed the size of the original file. The processor time to perform such an operation is also likely to be prohibitive.

It occasionally happens that a "compressed" file ends up being larger than the original. This occurs when there are few detectable areas of redundancy or garbage to remove. Then when the encryption map is added to the processed file, more total bytes are needed. This situation is most likely to arise with binary files like program files, where the distribution of bytes is sometimes fairly random, as far as most compression algorithms are concerned.

Once all noise and redundancy have been removed and the encryption map has been stored, there remains a core of information which cannot be compressed further. This is the information which is unique to the file, and distinguishes it from any other file. The file therefore cannot be further reduced in size without some loss of information. At this point, any further reduction in the compressed data portion of the file requires a corresponding increase in the encryption map. There are some absolute limits to compression. This is why additional compression cannot be achieved by running an efficient compression algorithm against a file multiple times.

Compression Methods

Turning now to the various methods of data compression, we find two general approaches, based upon whether the encryption map used is static or dynamic. The former uses the same map for all files, while the latter builds a unique map for each file based upon the information found as the file is processed. Static encryption information tends to be more compact and the programs more efficient since there are fewer variable factors to account for at run time. A large portion of the encryption map for all files can be kept as one centralized table, either in a database or as part of the program code itself. The disadvantage of this approach is that the amount of actual compression can suffer if there is too much variability between the files that one wishes to compress. A compression table that is optimized for COBOL source code is not likely to perform well when compressing a binary file.

Dynamic algorithms build compression tables at run time that are based upon the patterns actually found in the file currently being compressed. This generally results in a higher degree of compression across file types and contents because the pattern table is optimized for each individual file. The trade-off with this approach is that a unique pattern table must be generated for each file, which then becomes part of its encryption map. Depending upon the algorithm, this information can itself become rather large, defeating our original purpose. In addition, the run-time creation of a pattern table takes additional processor time, which can make these programs run more slowly.

We will now examine five compression methods which vary widely in technique. Several of these constitute the most common and successful methods currently used.

Repeating Character Packing

As mentioned earlier, most files contain a significant number of repeating character or word strings. In this method, also known as run length encoding, the file is copied serially, and each record is scanned for repeating strings. This is then replaced with some sort of flag to indicate a repeating string, the repeat character and a repeat count. In this way, the entire string can be represented by three or four bytes. Some algorithms can even consolidate this information to reduce the compressed image by another byte or two. This approach can generally reduce the size of ASCII files by 30 to 50 percent. Regardless of the compression algorithm used, the largest saving usually results from the removal of repeating strings.

Dictionary Look Up

If one can predict in advance which byte patterns will occur with frequency, those patterns can be pre-loaded into one centralized table. During compression of a file, patterns which are found in the table can be replaced with some sort of index to the corresponding table entry. This technique is similar in operation to the use of COBOL copy libraries, where predefined segments of source code are stored in a table and are referenced by a key value at compile time. This technique can be quite effective when many patterns are known in advance. This is often the case with ASCII text files like program source code, where particular textual words and word sequences occur with frequency. As file contents becomes more varied however, there will be fewer average "hits" in the table unless it is expanded to account for the contents of all file types to be processed. At some point this becomes self-defeating because not only will the table occupy much disc space, but the indexes into it also become correspondingly large, which diminishes the compression effect.

Huffmann Coding

In view of the foregoing, it would be desirable for an algorithm to generate a table that is sensitive to the patterns found in each individual file. One of the first such algorithms was developed a number of years ago and is called Huffmann coding, after its inventor. In its basic implementation, this method requires two reads through a file. During the first pass, the number of occurrences of each byte is tallied and a frequency table is built. Each byte is assigned a variable length bit string. Bytes which occur most often, like blanks and the letter "e" are assigned the shortest bit strings (this may be a single bit, in some implementations). On the second pass, each byte is replaced with its representative bit string. The table is then stored, along with the collection of bit sequences in the compressed file. As you might imagine, the double read through the file and the fact that this method does not operate on byte boundaries causes it to run painfully slow – particularly when implemented in software. Moreover, compression performance depends upon a highly skewed distribution of characters, so compression drops rapidly as a file's mix of characters becomes more even. Despite these limitations, Huffmann coding is still in fairly common use, especially in data communication equipment and facsimile transmission devices. Compression figures with this method are generally 30% to 40% unless the algorithm also scans for some common multi-byte patterns along with the individual characters. In this case, compression can be as high as 55%.

Adaptive Compression

Ideally, an algorithm would use a unique pattern table for each file, yet not need to store the table along with the compressed file. This has been achieved in a method called adaptive or LZW compression; a variation of a method originally described by Lempel and Ziv in a 1977 IEEE journal. This is probably the most successful compression technique to date. It requires just one pass through a file. A pattern table is progressively built as the file is processed. The algorithm takes one character and appends the next. If this string is found in the table, yet another character is added. When there is no "hit" in the table, this new string is added to the table. The compressed image is a table pointer with the novel character appended. Each compressed string is three bytes in length, and there is no need to store the table because it can be reconstructed at decompression time by a reverse process. Initially, compression is poor while the pattern table is relatively empty, but rapidly increases as the file is processed. This method can achieve compression ratios as high as 90%.

No single technique described above is the best in all situations. Therefore a sophisticated compression product will combine several of the above methods as well as other techniques, which may be file type or even hardware specific, in the effort to squeeze every last unnecessary bit out of a compressed file. While data compression is not appropriate for every application, approximately 30 years of research on the subject have minimized its limitations so that it has attained a significant place in the data processing industry for improving the efficiency of data transfer and storage.

Applications for HP3000 Users

A very common use of data compression is for archiving of source code. In Proactive Systems we have been using a Contributed Library product to do this for the last 6 years for COBOL and PASCAL source code and we typically save 75% of the storage requirement. We only use this for "archived" source, i.e. code not currently being worked on, but it only takes a few seconds to decompress a source file when needed.

Another frequent application is for archiving of data as a back-up method. For example one way to avoid the problem of staying overnight to mount "store" tapes is to simply copy a data base to another area of disc (e.g. to a private volume). This tends to only be practical if you compress the data at the same time as most people do not have enough spare space on their systems to hold their operational files twice over in full. Several third party software products provide this kind of capability, often combined with copying to tape (although the normal STORE program works fine on compressed files of course).

Accessing Compressed Files

However there are many cases where ad-hoc and frequent access to archived files is required. For example you may want to provide access to the last 5 years sales transactions for enquiry purposes. On a large data base it is impractical to decompress/recompress the files for every enquiry (you may not even have enough free disc space to hold them in uncompressed form) so what you really need is to be able to access the compressed data bases with your normal enquiry and reporting programs. We have recently developed a compression algorithm which enables you to do just that for IMAGE data bases.

You can certainly save a lot of disc space with this new solution (see the figures given above for IMAGE data bases) but what about the performance overhead? After all, we have to intercept the IMAGE calls (using the same technique as used in BACKBONE, OMNIDEX and SPEEDEX). Our initial figures indicate a small increase in cpu time because of the need to decompress each record as it is read. However you effectively read a third or less blocks of data so the I/O load is significantly reduced. For low volume, ad-hoc enquiry purposes, performance overhead is not likely to be a significant factor and the potential disc space savings are enormous.

The Ultimate Solution

Now obviously it is very useful to be able to keep data on-line in a compressed format and still be able to read it which is what we can achieve now. But an obvious extension of that is to provide update access. The major problems here are that not only can the compressed records "move" but during the compression process you have typically removed all the spare space in which to place new records! So it's not as easy as it may appear. In addition, on top of emulating all the IMAGE read intrinsics you have to emulate all the IMAGE writes as well so you effectively end up rewriting IMAGE. Needless to say, this is taking us a bit longer to produce but we know the practicality of it. With the additional read/write overhead we don't expect it to be the answer for all applications on all systems, but certainly any low volume ones will be suitable. With processor price/performance increasing more rapidly than the cost of disc storage, the benefits of data compression become more cost effective.

Computer Museum

Conclusion

I think you will see from what has been covered above that we see data compression as moving out of specialized areas of data processing and becoming a routine part of normal commercial applications. The technology has moved forward rapidly over the last few years so that very efficient, reliable compression algorithms are available. It's only a matter of applying the technology to the needs of users that remains.

The increasing number of applications for HPs DDS format tape drives

By Debbie Beech, Computer Peripherals Bristol.

TOPIC ABSTRACT

In January 1990, Hewlett-Packard introduced the first in a new range of secondary storage tape drives. The HP Series 6400 models 1300H and 1300S Digital Data Storage (DDS) format tape drives are based on Digital Audio Tape (DAT) technology. DAT is an innovative recording technology that offers a practical alternative to the longitudinal recording methods used in most existing storage devices. Significantly, DAT represents the first generation of helical scan devices designed exclusively for digital storage, bringing 1.3Gbytes of unattended backup on a single cassette.

DDS is a new technology for traditional applications. This paper looks at the features of DDS that make it suitable for these traditional applications and some newer applications also.

At introduction it was not possible to take full advantage of the drives' extensive range of features. One of these features is the Fastsearch capability, planned to be supported by the Summer of 1990.

In addition the Commercial Systems Group has announced a new version of the TurboSTORE software, TurboSTORE/XL II. This software package will dramatically increase the amount of unattended backup from 1.3 Gbytes on a single DAT cassette to over 18 Gbytes using data compression and four drives. Both of these enhancements will provide HPs customers, especially commercial system customers, a backup solution to meet their needs.

In summary the paper would describe the applications of HPs DDS format tape drives focussing especially on the enhancement Fastsearch and TurboSTORE/XL II have made to these applications.

INTRODUCTION

In January 1990 Hewlett-Packard introduced the Series 6400 models 1300H and 1300S DDS format tape drives. With this introduction HP became the first computer systems manufacturer to offer Digital Audio Tape (DAT) as a computer backup device. HP also offers DAT integrated into it's latest commercial systems, the HP 3000 series 922 and 932.

DAT has been the subject of significant press coverage and public discussion over the last two or three years. With a 1.3 gigabyte capacity it has captured the imagination of the major manufacturers in the tape drive industry and given tape a new lease of life.

At Hewlett-Packard we have been investigating helical scan technologies for a number of years. We looked into VHS and 8mm before deciding to develop DAT. We believe that DAT provides the best technology platform for a data storage medium to keep pace with disk backup trends through the 1990s.

DAT as a building block for data storage

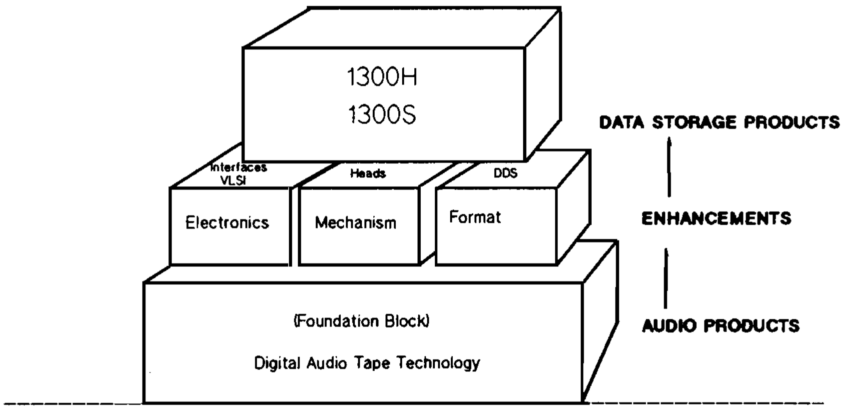


Figure 1. From audio to computer data storage

DAT was designed and developed by a consortium of 87 companies to provide a tape recording quality comparable to compact disk.

DAT technology offers many features that lend it to successful conversion to a computer peripheral product :

Firstly, the backing of a large number of manufacturers providing a "standardised" base technology.

Secondly, very high capacities at a low cost.

Thirdly, an ideal platform upon which to create an entire product family to meet backup needs throughout the decade. Form factor reductions and volume efficiencies will enable DAT to compete effectively with QIC and DC2000 technologies over the coming years.

Finally, a high level of data integrity. DAT mechanisms have been specifically designed for digital recording, unlike their VHS and 8mm predecessors which are analogue devices. Eliminating the need for a backward step to analogue techniques both simplifies the recording of the computer data and offers significant error rate advantages.

However, the storing of computer data is very different to storing music. The human ear is forgiving of a missing note in the middle of a symphony, but every bit is critical to a computer. Digital audio tape had to be modified to provide a product capable of data storage; a computer grade peripheral.

From an audio product to a digital data storage product

The key to storing digital data on a DAT drive is the "Digital Data Storage" format. The DDS format takes the audio format as its base and adds extra "layers" of error correction and management as well as additional organisational features to facilitate access to the 1.3 gigabytes of data.

A great deal of research was necessary to ensure that the enhancements to DAT through the DDS format were based on potential users applications. This research showed that the traditional tape applications were still of paramount importance to the majority of end users. Backup, increasingly unattended backup, data interchange and software distribution were cited as requirements in the high-end PC, workstation and multi-user systems markets. These requirements along with requests for compatibility with existing software and a standard recording format, provided the goals for designing a recording format to support computer applications.

The following section outlines the applications of the DDS tape drives for storing digital data with HPs computer systems. Particular emphasis is give to the newly supported fastsearch feature and the introduction of the TurboSTORE/XL II software.

APPLICATIONS OF DDS PRODUCTS

BACKUP

This is the primary application of a tape drive for most users. As disk capacities grow to a gigabyte and over, operator time in backing up these disks has also increased. The industry emphasis on "unattended backup" becomes more important than ever with the trend towards a "lights out" computer room.

Unattended backup

Backup performance is measured by the rate of data transfer combined with the amount of operator intervention. A DDS product allows 1.3 gigabytes of user data to be backed up in under two hours with no operator intervention, resulting in quick, cost effective and reliable backups.

TurboSTORE/XL II

TurboSTORE/XL II was introduced on July 1st 1990. This software, available for the HP 9000 Series 900 family, has received dramatic new capabilities, including :

* Data compression that triples the performance of the DDS drive
* Unattended backup of up to 17 Gbytes
* Optional on-line backup
* Mirrored disks

By combining these capabilities with the DDS format tape drives a threefold improvement in the potential backing storage capacity and performance can be achieved.

TurboSTORE/XL II software data compression increases data backup performance, particularly for the high-end Series 900 systems with the more powerful CPUs. The data compression algorithm used can provide either 1.8:1 or 3.6:1 data compression, thus providing the potential for storing data up to 3.6 times faster. Using a Series 960 a performance of 2.1 gigabytes an hour can be achieved with a DDS tape drive using the 3.6:1 data compression algorithm. A total of 4.3 gigabytes of data can be stored on one DDS cassette in just over two hours. TurboSTORE/XL II supports up to eight backup devices running in parallel, so overcoming the performance bottlenecks.

Furthermore TurboSTORE/XL II eliminates the unproductive system idle time spent waiting for cassettes to be rewound or changed during backup. This is done by storing sequentially to multiple DDS tape drives. After one cassette has been filled the backup automatically switches to filling another cassette on the next drive while the previous cassette is simultaneously rewound.

The combination of TurboSTORE/XL II data compression with multiple DDS tape drives will give true unattended backup of low-end and mid-range systems. With four DDS drives and a 3.6:1 data compression ratio unattended backup of over 17 gigabytes can be achieved.

For environments that require 24 hour a day system availability, TurboSTORE/XL II offers and optional on-line data backup capability. On-line backup supports round-the-clock transaction processing without sacrificing data backups or requiring redundant mirrored disk drives.

Cost of Backup

By using DDS drives for backup the cost of operator intervention time can be significantly reduced. This cost of backup is reduced even more by the low cost of the drive and the media. The DDS drive itself is priced at roughly one-third the price of HP's 6250 bpi tape drive, and approximately half of the 1600 bpi. Further savings in the cost of media - half the price of a 1/4-inch cartridge, significantly impact the cost of backup especially when calculated over an extended period of time. You can see from the following chart a comparison of the number of pieces of media required to backup over a gigabyte of data.

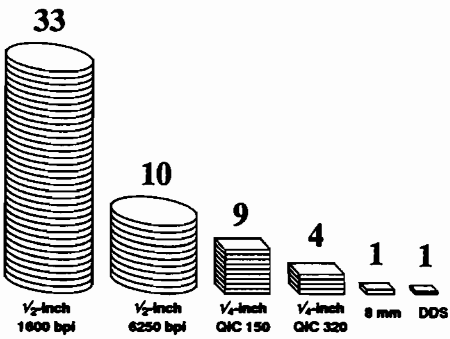


Figure 2. A comparison of the number of pieces of media used to store 1.3 Gigabytes of data

Performance

Although unattended backup is the key requirement of this application, performance of a product must be sufficient to ensure that backup does not interfere with normal production cycles and system up-time. DDS products have a transfer rate of 11 megabytes/minute, approximately equivalent to the performance of a 1600 bpi tape drive. What is more, the time taken for an operator to load and change tapes may be eliminated by the large capacity of a single DDS cassette. This is clearly demonstrated by a comparison of the time taken to backup 600 megabytes using the different technologies.

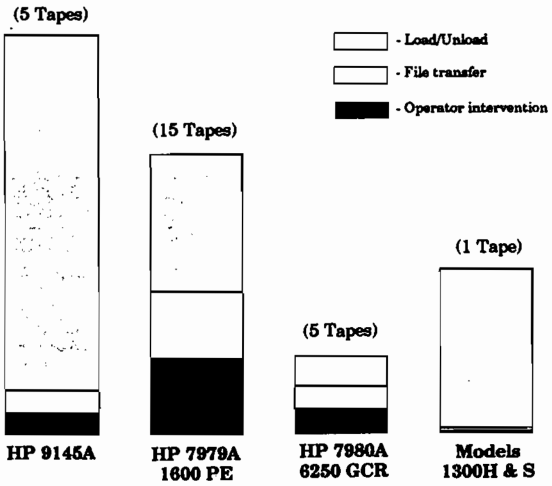


Figure 3. Time to backup a typical system of 600 Mbytes

Ease of use

Ease of use is an important feature in making backup as painless as possible. With a load time of 20 seconds there is a marked difference between DDS and traditional tape drives. Furthermore DDS cassettes do not require pre-formatting.

Reliability

Reliability advantages result both from DAT technology and the DDS format. DAT offers reliability advantages common to helical scan technology including low tape speed for reliable handling. Other features specific to DAT make the tape handling even more gentle by using electronic rather than mechanical tensioning. This reduces repositioning time and further improves reliability by reducing peak tension on the tape during repositioning. Furthermore DAT contains half the number of parts used by conventional helical scan devices.

DDS has been designed to give high levels of reliability and data integrity. A total of ten error avoidance techniques have been used the main ones being three levels of error correction and read-after-write. Error monitoring and a high level of media specification add to the reliability of these products.

ARCHIVAL

Most companies house rooms full of archived data - a costly and inefficient use of resources. The cost of storing data for any length of time is expensive in terms of space and media.

Cost of data storage

The inexpensive DDS media makes it a very cost effective means of storing large quantities of data - especially in terms of the cost per megabyte. The cost saving incurred when storing the credit-card sized media also make it more attractive than traditional tape library media, especially in applications where duplicate copies of files need to be stored off-site. Imagine storing 10 gigabytes to 10 DDS cassettes and then imagine how many 1/2-inch tapes you would require to store the same amount.

Many companies own large fire and bomb proof data chambers just to store their tape libraries. The size of DDS cassettes enables the use of a standard fire-proof safe freeing up the chamber.

HP's environmental testing makes us confident of a media life of 10 years or more under typical archival conditions. What is more during this time the cassette will not require the usual operator intensive re-tensioning. An error monitor and media log add to the backup reliability by informing the user when a tape requires replacing or cleaning.

Fastsearch

With 1.3 gigabytes of data on a cassette it is important that access to a file can be performed quickly and easily. Fastsearch is a new feature for tape drives that is unique to DDS. Earlier technologies take several minutes to locate a file, however DDS has an average access time to any file on the tape of 20 seconds, 200 times the normal read/write speed.

The structure of the DDS format has been perfectly matched with the capabilities of DAT for file storage and retrieval. Because DDS stores files contiguously and sequentially an entire file will be accessed in a single fast search motion. This maximises the random access performance of DAT as well as improving the drive reliability.

This feature is supported on the following operating systems:

HP 3000	MPE/XL - 2.1
	MPE/VE - No support planned
HP 9000	S300 - HP-UX 8.0
	S800 - HP-UX 8.0

DATA INTERCHANGE

Data interchange is an every day application for many tape drives. Sending data around the company or even to external companies can be complicated by ensuring that the right media is used, a certain format adhered to or that a particular system is used to retrieve the data.

Standards

The need for a standard format combined with the other customer requirements for storage devices influenced the development of the DDS format. A large group of manufacturers are currently supporting DDS as the single format standard for DAT including over 70% of the world's 1/4-inch tape drive manufacturers. This co-operation with the format development has led to the submission of DDS to the American National Standards Institute (ANSI) and further submission to the European Computer Manufacturers Association and the International Standards Organisation (ISO)

With the backing of the standards organisations and the industry manufacturers DDS has the potential to become a universal standard for data interchange in the same way as 1/2-inch reel tape.

DDS MANUFACTURERS GROUP

1/4" Company	1/2" Company	8mm Company	Plus
Archive *	LMSI	Exabyte	Sony *
Wangtek *	Cipher		Alps *
Cipher	Hewlett-Packard		Alliance
Hewlett-Packard *			Wangdat *
			Mitsumi *
			M4 Data
			Aiwa
This represents **72%** of 1/4" drives sold in 1988	This represents **40%** of 1/2" drives sold in 1988	**100%** of 8mm drives sold	+3 non public licencees

* Publically announced DDS format tape drives

Figure 4. Industry backing for the DDS format

Digital Equipment Corporation have announced support for the DDS format and will be introducing a product with their workstations this summer.

HP System support and interchange

As DDS is supported on all HP 3000 systems data can be interchanged easily between MPE/VE and MPE-XL systems using a single cassette. This is important for migration purposes. Furthermore as DDS is also supported on HP-UX, data can be swapped between HP 3000 multi-user systems and workstations. For computers that are not connected electronically, DDS provides low-cost, high data integrity and a standard recording format to create an alternative to floppy disks and other tapes.

SOFTWARE DISTRIBUTION

The combination of a low cost, compact, robust, high capacity media means that a cost effective software distribution medium is available using the same device as is used for backup.

Commercial distribution

Value-added-resellers in particular will benefit from the savings made from software distribution using the low cost DDS media, both from cassette outlay and postage cost. The media is both robust and light-weight.

A process known as "contact printing" can be used with DDS to provide a reproduction method for fast and reliable software duplication at low cost. This process will become more prominent as DAT takes off in the audio world and music is reproduced in large quantities.

HP's software distribution strategy

Software for HP's 3000 systems has been available on DDS cassettes since introduction. For HP 9000 systems, the Series 832 is the only system that currently supports software distribution on DDS cassette. With the release of HP-UX 8.0 - DDS software distribution will be supported across the range of Series 800 systems. We are also working to provide DDS software distribution on Series 300's and hope to announce support shortly after the release of HP-UX 8.0.

With the high capacity feature of the technology the number of cassettes required to hold an operating system release will be significantly reduced.

BOOT AND INSTALL OF A SYSTEM

Although boot from DDS has been available for HP 3000 systems from introduction, this capability has not been available for the HP 9000 Series. From HP-UX 8.0 HP 9000 systems (both Series 300 and Series 800) will also have Boot capability.

POSITIONING

As you can see, DDS is suited to the traditional tape drive applications currently covered by 1/4-inch and 1/2-inch tape drives. At Hewlett-Packard we do not see DDS fulfilling all the needs of our customers in every application. We expect that DDS will complement the current tape drive product lines to provide a complete range of solutions.

Positioning DDS with 1/2-inch and 1/4-inch tapes

The 6250 bpi tape drives have twice the backup performance of a DDS drive. If high speed file transfer is more important to you than high capacity unattended backup then the 6250 bpi is still the best match for your needs. Furthermore 1/2-inch reel-to-reel tapes are an established industry interchange standard for computer data. HP's goal is to establish DDS as a future interchange standard.

The 1/4-inch cartridge tape drive continues to be a good match for the low-end disks and for customers requiring an entry level tape drive. These drives remain the lowest cost tape drive solution offered by HP.

DDS is the best match to your needs if you require unattended backup of capacities of 300 megabytes or more, where you wish to reduce operator intervention. DDS is also a much lower cost solution than 1/2-inch tape drives in terms of both the drive and the media. Furthermore DDS brings you savings in floor space.

The bottom line is that we believe DDS to be the best backup technology for requirements positioned between 1/4-inch cartridge and 1/2-inch reel to reel tape.

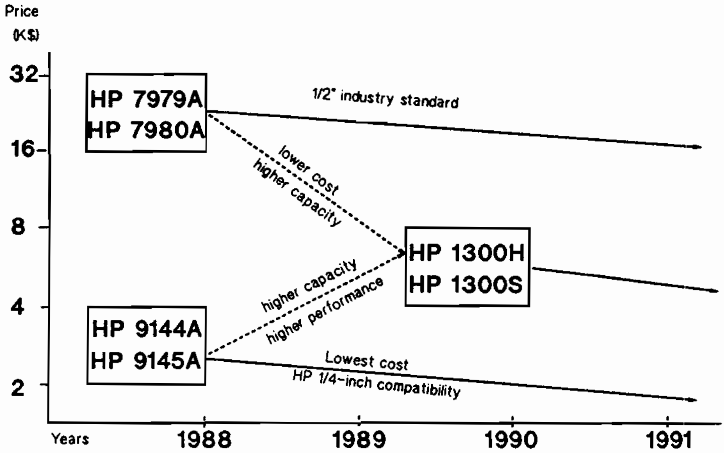


Figure 5. DDS positioned with traditional tape drives

DDS positioned with optical disk drives

HP has invested heavily in both DAT and rewritable optical technologies. We believe that together these two technologies address a wide segment of the market, while DDS addresses a traditional market segment, rewritable optical addresses a new market segment. As such the two can coexist.

Rewritable optical fits into a gap between primary and secondary storage - Direct Access Secondary Storage (DASS), a gap containing previously omitted or poorly met user needs. It offers direct access to files traditionally kept off-line. Trends in information management suggest a need to keep more information online that is possible today, and rewritable optical provides an economic alternative to conventional on-line storage technologies.

Even with it's high speed search capability, DDS cannot be positioned as a random access device. Average access times are 300 times lower than an optical disk drive. The appeal of DDS to the majority of users remains as a high performance, unattended, cost-effective backup device.

THE FUTURE OF DDS

HP has just introduced it's first generation of DDS product. This is by no means the last you will hear of the technology. The continued growth in disk drive capacities calls for constant innovation to keep up the pace in the secondary storage market. One of the key reasons why HP choose DAT technology was its ability to provide a platform for several generations of products each providing complete backward and forward compatibility with the previous one.

We see the potential for moving from todays product to future generation products offering over 10 gigabytes of capacity per cartridge and with transfer rates at over 2 megabytes per second. These improvements could be achieved without the need for a new format and could include : longer length tape, faster mechanisms, data compression and thinner tracks on a tape. Because the DDS cartridge is small and the mechanics of the drive are relatively simple, we also see significant opportunity for the creation of drives that are smaller and less expensive.

CONCLUSION

HP has introduced the DDS format tape drives to meet the needs of your traditional tape applications: backup, archival, data interchange and software distribution. In each of these areas DDS can add significant value reducing the cost of backup, operator time and storage requirements.

DDS is not positioned as meeting the needs of every application and will coexist with the other technologies in HP's secondary storage product line. What is more DDS and rewritable optical form a complementary set of products.

Over the next few years you can expect to see increased capacities, performance, lower cost and smaller form factors with HP's DDS products. HP is committed to this technology in order to meet the needs of your traditional tape drive applications today and into the future.

Optical Storage Benefits, Applications, and Solutions

John Szlendak and Anne Gleason
Hewlett-Packard Company
Greeley Storage Division
700 71st Avenue
Greeley, Colorado 80634

Imagine having a computer system with enough storage capacity to save any data, text, image, graphic or voice file without worrying about having enough disk space. Imagine being able to retrieve a file that you haven't accessed in years, within seconds, without having to search for a tape buried in a tape library. Imagine feeling secure, knowing that your files, backed up automatically, will last for a minimum of 10 years.

Rewritable-Optical products offer reliable, economical, easily accessible high-capacity storage, which was previously unattainable with magnetic disk and tape. Rewritable Optical storage is the solution that can make the dream system come true.

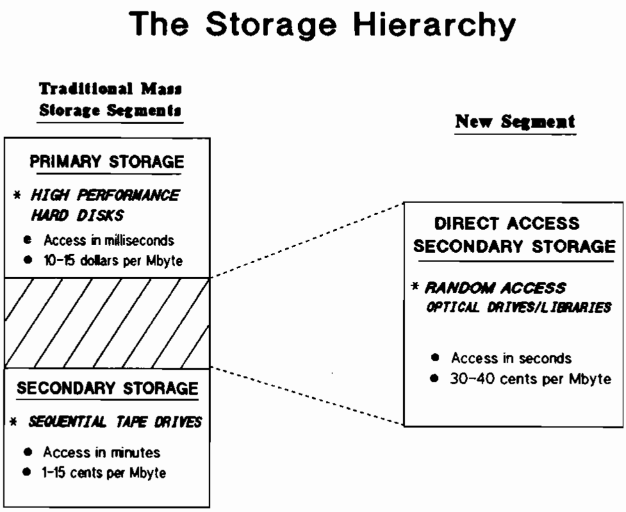


Figure 1

These new capabilities are possible with Direct Access Secondary Storage, or DASS. A new kind of storage, made possible with Rewritable Optical disk technology. Traditionally, mass storage solutions have fallen into one of two categories - Primary and Secondary Storage. Primary storage is typically one or more fixed magnetic hard disks, with fast, random access high-capacity storage. Secondary storage has consisted of one or more offline storage devices - usually a 1/4" or 1/2" tape drive, with slower performance and lower storage cost. This leaves a large gap in price and performance between these two categories (Figure 1). Rewritable Optical bridges this gap by offering economical online access to off-line information. With an average access time of 0.1 second for the drives and 10 seconds for library systems, and with the media price of less than $.40 per megabyte, optical technology creates a new layer of storage in the mass storage hierarchy.

Studies of hard disk storage have shown several trends:

* A large portion of data stored on hard disks is static - a great deal of data isn't used for long periods of time.

* Users are reluctant to remove data from their disks because of the inconvenience of selecting the files to be removed, and the added inconvenience of retrieving the files from tape.

Rewritable Optical libraries will provide a solution to these inconveniences:

* Because of the tremendous cost advantage of Rewritable Optical disks over hard disks (cents versus dollars per megabyte), data can be kept in a library where it will be inexpensive, yet still easy to retrieve.

* The job of managing hard disk space can be automated - older, static files can be migrated to inexpensive storage while the system is running.

* Using file server technology, these benefits can be brought to PC and workstation networks.

The Rewritable Optical disk library system can make these applications transparent to the user. Appropriate file management software can track files on many disks and swap disks automatically, bringing vast amounts of unattended storage online without operator intervention.

Whether your storage need is large enough to require a multi-disk library system or is adequately served by a standalone optical drive, the applications are basically the same. Direct Access Secondary Storage offers exciting new possibilities in three major areas - online archival storage, unattended backup, and electronic image management.

Online Archival Storage

Rewritable Optical disks are convenient and reliable for archival data storage. Whether data is archived for a week or ten years, it can be accessed and updated easily because data can be directly written to and read from the optical disk. From the users' point of view, archived or historical data is as accessible as if it were still on the primary storage media - with somewhat slower response times.

The computer industry is evolving from host configurations towards network configurations (Local Area Network) in which work groups share resources. For instance, it is common for a workstation user to be connected to the server through a LAN. The server is based on a hierarchical storage architecture that integrates optical storage as a transparent secondary storage device for magnetic disk drives. Magnetic files are automatically migrated to the optical disk drive to free up Winchester storage as needed. The migration time can be based on how much time has passed since the file was last accessed, the size of the file, and so on. When the user requests a file which has been migrated, it is retrieved from the rewritable optical disk, and migrated back to the magnetic disk, without user intervention.

The benefits of archival applications are clear: Rewritable optical products bring historical information online and facilitates better data security and peace of mind for users who require more timely access to their information.

Unattended Backup

System backup and recovery can be fully automated with an optical library. Automated backup ensures that backup occurs and is completed as scheduled, rather than at the operator's convenience. Data security can be enhanced by holding the backup media within the library, thereby reducing the risk of of lost or misplaced cartridges. Automated backup also frees the operator or system manager to complete more important tasks that, in most cases, can pay for the optical library system itself.

The optical library can be used with TurboSTORE on the HP3000 series 900 to achieve unattended backup. It provides a superset of the features provided by the Store/Restore facility. TurboSTORE supports both consecutive and concurrent backup devices for enhanced unattended backup performance. It manages media swaps and spreads files across optical devices. TurboSTORE provides software data compression and software online backup. This boosts optical's performance and virtually eliminates backup downtime by running it concurrently.

HP-UX users can use the library system to centrally back up networks of workstations or PCs overnight, between shifts, or as a continuous background utility. Users can write simple scripts to automate the process, or take advantage of some of the more sophisticated backup utilities on the market. These backup utilities let the user backup raw partitions, special device files, and files larger than a single backup volume. The user decides exactly what part of each file system to back up and when. Some utilities create an online catalog which shows the storage location, owner, permissions, file size, and the date the file was last modified. The user can then request the file that has been backed up, and receive it without any operator intervention.

Electronic Image Management

Stored images are a valuable information resource for many businesses. Image (bit-mapped) storage requires a lot of space. A single 8x10 page of scanned (300 dpi) information can occupy over 1.0 Mbyte of storage space. That is why image/document data has traditionally been kept offline on paper or microfilm. With optical storage, this data can be stored online and electronically managed and shared among many people.

As an example, some engineering management software allows vast numbers of scanned engineering drawings to be stored online with the optical library system. Not only are storage space and labor costs reduced, but the data is directly accessible for retrieval and electronic distribution. These drawings can include hand-drawn originals, blueprints, aperture cards and other forms which are handled with speed and economy. This type of solution is available today from GTX Corporation.

Hewlett Packard is also offering an optical disk based Image Management solution that captures and manipulates hardcopy information (images) as part of an integral business information system. HP's AIMS (Advanced Image Management System) can put an end to the paper problem by bringing information such as handwritten letters, photographs and forms online. Such information can be duplicated electronically and stored with other electronic forms of data like text and graphics.

Rewritable optical technology can be used in a wide variety of applications especially those requiring large amounts of storage. No longer will high storage cost be the limiter in improving information access, management and opperational efficiency. It allows you to have low-cost, online access to information that makes your business more competitive.

Cost Justification

In order to determine the value of optical storage one must first look the challenges facing information management and the accompanying business basics:

Challenges:		Solution:
* Online Access Needs Increasing	-	Look for lower cost storage medium
* Historical Data Needs Increasing	-	Look for ways to bring off-line information on-line.
* Operating budgets decreasing	-	Look for ways to cut costs, automate and increase productivity.
* Customer service & satisfaction	-	Provide timely and accurate data
* New information types (ie. image)	-	Innovate

These value parameters can be summarized as Storage Cost, Operating Costs, and Information Time Value.

Storage Cost Comparisons

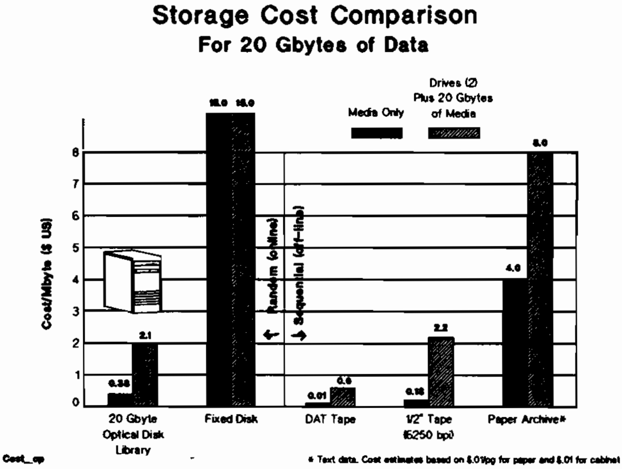


Figure 2

Given the choice, all information would be kept online. But looking at the $/Mbyte of storage comparison in Figure 2, it might not make economic sense to keep it all on a hard disk. Because of this, less than 1% of all of today's business information is kept online. Optical storage, and an optical disk library system in particular, can certainly change that. In the above graph, the cost of a 20 Gbyte optical storage system is compared to that of DAT (Digital Audio Tape), 1/2" tape, hard disks, and paper. Optical storage is significantly less expensive than hard disk storage. And while today's optical media cost is twice that of 1/2" RR tape ($/Mbyte), the overall cost, including two drives for storing and retrieving 20 Gbytes of data is about the same.

DAT technology provides the lowest cost per Mbyte of storage, but unlike optical it is not random access or online accessible.

While the cost of paper archive is perceived low, the overall cost of paper, cabinets, and floorspace for large volumes makes it more expensive on a cost per Mbyte basis than electronic storage on optical media or high capacity tape.

Operating Costs

The most significant part of the operating expense is human resource. Much of it involves backup and archive operations. With the shrinking operating budgets, labor intensive backup practices are less desirable. "Unattended", "Lights Out" and "Operatorless" are the new industry buzz words.

Since optical disk library systems can automate the whole backup and archive process, they can significantly increase its frequency and reduce the operating cost. The total saving, depends on how labor intensive the operation. A good estimate to is $30,000/year per operator. There's also also the floorspace cost

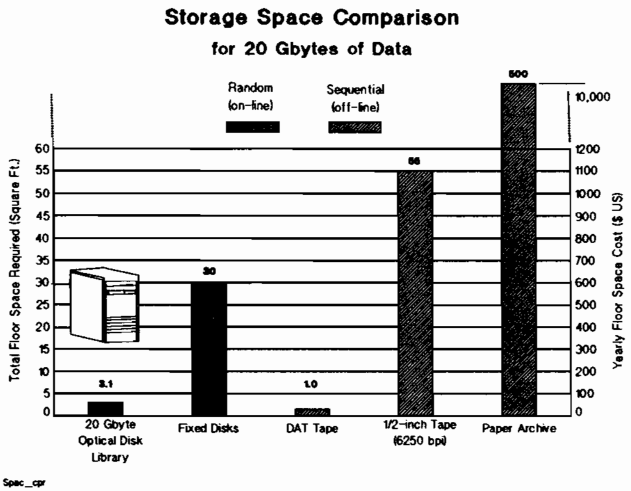


Figure 3

Figure 3 compares the storage space requirements for 20 Gbytes of information. Next to DAT, the optical disk library system is the most compact storage solution of the alternatives above. A 20 Gbyte system can fit in slightly more than 3 square feet of floorspace.

Information Time Value

The most difficult to measure, and probably the most important value parameter, is the accuracy and time value of information. Competitive advantage is gained through timely decision making based current and accurate information. Each company can place a different value on this. Some can afford to keep historic information online, others can only keep it on tape, microfilm or paper. The time value of this information is directly proportionate to its storage cost.

Studies estimate that 1.3 trillion paper-based documents are currently stored in US offices (excluding government) and it is doubling every four years. The US Archives and Records Service also states that 3% of these documents were misfiled. The cost of retrieving just one of them is estimated at $120.

Saving time directly translates to better service and customer satisfaction. It also means higher productivity. BIS International estimates that the benfits of an electronic imaging system that brings offline information online are:

Transaction volume increases	30% +	Space reduction	50% +
Staff reallocation	30% +	Transaction time reduction	50% +

A Return Analysis Example:

A small manufacturing company producing durable goods accumulated quite a bit of information in its 10 years of business. It has 6 Gbytes of data on-line, 30 Gbytes offline in the tape library (700 1/2" tapes) and much more in paper archives.

Running out of space and realizing the difficulties of managing and accessing this information, the company invested $40,000 in a 20 Gbyte Rewritable Optical Library System. Not only did it automate their backup operations, it also provided them with online access to much of their offline data.

The company cut expenses right away by automating their backup process with the optical disk libray and replacing the need for one of their operators. They also placed 30% of their 1/2-inch tape data on optical for easy access and migrated 20% of their least frequently accessed hard disk data to it for lower cost. The total savings they realized was:

Operator savings ($30K/year)	$ 30,000
Hard disk space saving ($15/Mbyte)	18,000
Tape saving ($18/Reel)	3,780
--------------------------------	------
Total Cost Saving	$ 51,750
Optical Disk Library Cost	40,000
--------------------------------	------
Total Benefit after a year	$ 11,750

In this case, the savings they realized in managing their computer data easily payed for the optical library system intself in less than a year.

Based on this success, the company is now lloking at another system to manage their vast paper archives for easy access and floor space savings.

Summary

Rewritable optical storage provides many new capabilities not previously possible, especially in online archive, unattended backup and image management.

Cost/Benefit

Rewritable Optical Disk Library System

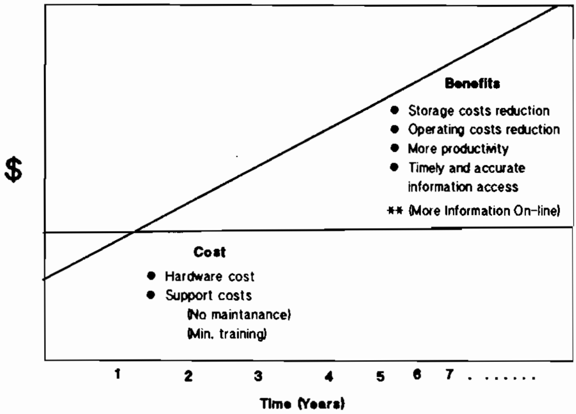


Figure 4

The benefits gained through increased productivity and cost reductions can easily pay for themselves in a very short time. This does not even include the most important benefit of all, the ability to have easy access to more information online.

Mass Storage Technology Positioning

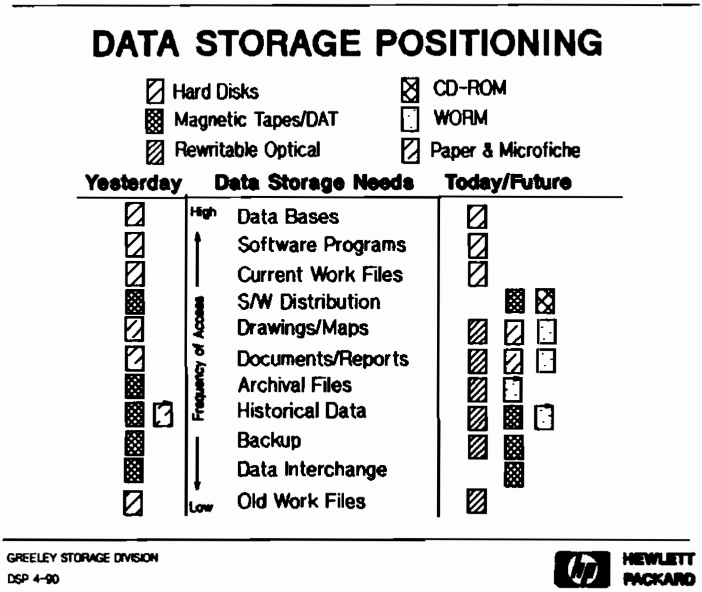
Jon Schiedel

Hewlett Packard Company
700 71st Ave.
Greeley,CO
80634

(303) 350-4385

Introduction:

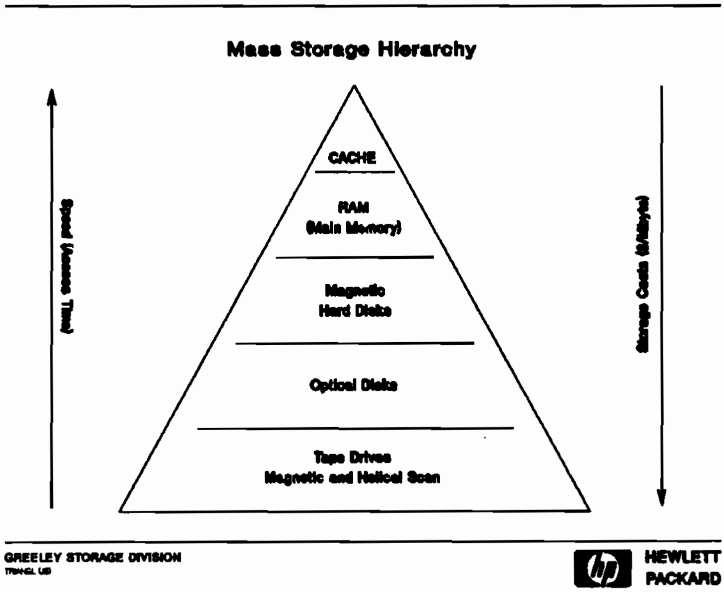
Recently, mass storage has gone through many changes. The development of new mass storage technologies as well as improvements in performance and reliability of older technologies has made positioning of the technologies as clear as mud. It wasn't long ago that you based your secondary storage buying decision on price and performance alone, but the recent development in advanced data storage technologies makes it much more complex. The broad spectrum of technology offerings available today makes it confusing as to which is best suited to meet your needs, especially in the area of SECONDARY STORAGE. Each of the technologies has its own unique qualities as you will see from this presentation. Today I will be talking about secondary mass storage positioning because this is where most of the technological changes are taking place.



Although the new technologies overlap older ones, none of the newer ones replace the old ones. So how do you decide which are the right ones for you? First you must understand what the technologies are, how they relate to one another, and then decide which ones are best suited to your applications.

I. Memory Systems Hierarchy

As shown in this slide, high speed access to data is costly. The most expensive storage is CACHE memory, the least expensive is magnetic and helical scan tape systems.



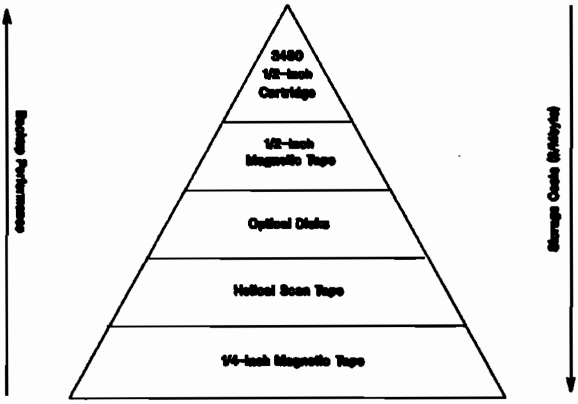
A. Primary Mass Storage:

The roll of primary mass storage hasn't changed much since it was first developed. Of all the mass storage technologies available, magnetic hard disks still offer the highest performance. The hard disk price/performance still prevents it from expanding into the secondary mass storage area.

B. Secondary Mass Storage:

Secondary mass storage technologies vary in price and performance as shown in this next slide. The highest performance is provided by 1/2-inch magnetic tape followed by optical disks, then helical scan tape and lastly 1/4-inch cartridge tape, but price and performance cannot be the only factors to consider when buying secondary storage.

Secondary Mass Storage Hierarchy



GREELEY STORAGE DIVISION

HEWLETT PACKARD

Secondary Mass Storage Applications

Before you begin to decide which secondary mass storage product will meet your needs, first you must identify for what applications the mass storage will be used. The traditional secondary mass storage applications are presented below.

Secondary Mass Storage Applications

- Backup
  - Full Backup
  - Partial Backup
  - Data Logging
- Data Interchange
- Software Distribution
- Archival Storage

GREELEY STORAGE DIVISION

HEWLETT PACKARD

Backup

This is the primary application for secondary mass storage. To protect your data if there is a disaster, such as a disk failure, fire, flood, etc. The three backup methods most used today are full backups, partial backups and data logging. Full backups are typically performed once a week and stored off site for a period of time. Partial backups are normally done each day (sometimes twice a day) to back up files that have been altered since the previous e partial. Data logging is typically done when altering data bases. When logging to tape and/or disk, each keystroke is saved. It is not uncommon for all three methods to be used on the same system.

Data Interchange

GREELEY STORAGE DIVISION

HEWLETT PACKARD

Data Interchange

When you want to move your data from from one HP system to another or to a non-HP system you require removable media and a standard technology that is widely accepted and supported across the computer industry.

Archival Storage

Certain data must be stored off site for very long periods of time. Good examples of this are legal, tax, and payroll records to name a few.

Software Distribution

Is very similar to Data Interchange in that it requires a standard technology that is widely supported accross the computer industry.

New Technologies and New Applications

The development of the new high capacity optical disk and helical scan technologies has expanded some of the applications so you can save more time and money. In order to realize these savings, new software solutions are required.

New Secondary Mass Storage Applications

- **Unattended Backup**
- **Online Archiving**
- **Image Management**

GREELEY STORAGE DIVISION

HEWLETT PACKARD

The applications that I am referring to are:

Unattended Backup

Doing a backup without operator intervention gives you the the ability to start your backup before leaving the workplace for the day. Someone would return at a later time, perhaps in the morning, to eject the media and store it in a safe location. This may be acceptable to some, but others will not leave the backed up data on site for a prolonged period of time because of the risk of a possible disaster (fire, flood, disk crash, etc. After all, the reason you ran the backup was to ensure that there would be no risk of data loss.

Online Archiving

Requires special software which automatically writes data to a backup media files which have not been accessed for a pre-determined amount of time. This data will remain accessible (on line) without the need of an operator. By automatically moving the files off the magnetic media, you will be stretching your magnetic disk space.

Image Management

Large, complex, computer generated drawings or scanned documents which normally would be stored offline can now be kept online and accessible with rewritable optical disks.

II. Technologies and their Applications

Random Access Secondary Storage

Magneto Optic or rewritable optical disks offer large capacity plus random access performance. These features make MO well suited for backup, either unattended or online archiving. The optical disk technology is also ideal for document storage and retrieval. Data normally stored offline which you may need to periodically access can be easily kept online with an optical disk library.

MO performance today is comparable to a 6250 bpi tape drive and the media is rewritable without requiring replacement (it doesn't wear out), so the cost of about $250 for each cartridge is easily justified.

WORM offers lower read/write performance than MO, but it is better suited to storage of legal and financial documents. The media is approximately 1/2 the price of MO media, but it can be used only once.

CD-ROM is not written to by the user's computer. data must be prepared and mastered using an expensive specialized publishing system. Subsequent copies of the disk can be inexpensively duplicated from the master. CD-ROM is best suited for distribution of publications, training packages, catalogs, and of course, software.

Serial Access Secondary Storage

Magnetic Tape

1/4-inch cartridge tape technology was first introduced in the early 1980s. There has never been a completely compatible industry standard interchange format developed for the 1/4-inch tape, and because of this, it never became suitable for data interchange. Although the capacity per tape cartridge is quite high, its performance limits its positioning as an entry level backup solution.

1/2-inch open reelis the oldest of all the secondary mass storage technologies covered here today. 1/2-inch magnetic tape offers three industry standard interchange formats, 800 bpi NRZI, 1600 bpi PE, and 6250 bpi GCR. 1600 bpi PE format is the most widely used of the formats for what we call mid-range backup (100 to 500 Mbytes), data interchange, and software distribution. The higher density GCR format which writes dat at 6250 bpi density is used mainly for high speed backup for mid-range to high-end systems. 6250 is also used heavily for archiving data for long periods of time.

1/2-inch cartridge (3480/3490) tape drives offer very high performance (about 8 times that of a 6250 bpi, 1/2-inch tape drive), but it is too expensive for most HP customers (in the neighborhood of $90,000 each), and because of its expense is used mainly in mainframe environment.

Helical Scan

DAT - Of the helical scan technologies, the DAT DDS format has the widest support in the industry for establishment of an industry standard interchange format. Once the ANSI standard has been established and this technology is widely used in the computer industry, it will be ideal for data interchange and software distribution. At this time, DAT is positioned as an unattended backup solution for up to 1300 mbytes of data. Its performance slightly exceeds that of a 1600 bpi, 1/2-inch tape drive, so it is well suited for backing up low-to-mid-range systems. The DAT tape cartridge is quite inexpensive at approximately $15 each.

8mm helical scan technology was introduced several years before DAT. It is fairly widely used in the computer industry, but hasn't gained the backing as an industry standard interchange format like DAT DAT DDS format has. Storage capacity per cartridge is is higher than DAT, and cartridges are priced in area of $8 to $10 each. Performance of 8mm is about 1/3 higher than DAT and 1600 bpi, 1/2-inch tape drives and is positioned as an unattended backup solution for mid-range-to-high end systems.

III. Other Positioning Influences

Technological Advances

With hardware data compression, backup performance and media capacity has increased enormously. When adding DC to the new, higher capacity technologies,

Performance Prediction Techniques

Dan Sternadel - Tim Twietmeyer

Hewlett-Packard Performance Technology Center

8050 foothills Blvd

Roseville,CA 95678

Overview

This paper gives an overview of the complex discipline of capacity planning. It is intended for system managers who want to spend more time proactively managing their systems and less time troubleshooting problems and responding to complaints.

It is also intended for business managers: those people not directly involved with data processing departments, but whose business has become increasingly dependent on computers. For these people, this paper serves as an introduction to how capacity planning can save time, trouble, and money.

What Is Capacity Planning?

Capacity planning means ensuring a future environment in which a target level of service is delivered and maintained at a reasonable cost. It means having the right resources at the right time to satisfy performance expectations.

Critical to this definition is the word *future*. Other areas of performance management—diagnosis, system management and operations, application optimization—are used to analyze current resource requirements. Only capacity planning expressly addresses the future.

Why Do Capacity Planning?

Dependence on the computer translates into ever-increasing portions of a company's budget going to data processing. As such, there is growing concern for system reliability, availability, responsiveness, life cycle, serviceability, and cost. No longer are these concerns the province of data processing departments alone; increasingly, business managers must get involved.

How Do You Plan Capacity?

Capacity planning—like all planning—requires knowing where you are today, where you want to be tomorrow, and how to get there. This translates into taking the following steps:

- Understand today's data processing and business environments.
- Predict changes in the workload and the business.
- Set performance objectives.
- Model to predict performance.
- Evaluate alternatives.
- Maintain capacity planning information.

These apparently simple steps are discussed in greater detail in the rest of this paper. It should become clear that each step requires care and skill to execute properly.

Understanding Environments

A good capacity planner must have one foot in the data processing world and the other in the business world. Companies cannot afford data processing departments that function in isolation from other departments, especially those tasked with business planning.

The first step in capacity planning is to gather information about both the data processing and business environments, including the following:

- System status.
- System workload and performance.
- User service level expectations.
- Business plans.

System Status

A system is statused by taking an inventory of all hardware and software on it. This includes assessing system capacity, which is simply the maximum amount of work that it can do in a given period of time.

This inventory provides a current picture of the system. It forms the basis for planning system changes and assessing their appropriateness.

System Workload and Performance

It is important to know what work the system is doing and how well it is doing it; that is, how well it performs its job.

System performance is necessarily expressed in terms of workload, which is how much work it does in a given amount of time.

Workloads are expressed in terms of workunits, which are fixed quantities of work that the computer system does. A workunit might be one batch job or one command. Thus a workload might be 100 batch jobs per hour or 10 transactions per second. Clearly, several types of workunits may be applicable for any system.

What complicates this simple concept is the complexity of both the system and its workload. The workload is comprised of many components, each impacting system resources (CPU, I/O, memory, disk space) differently. This leads to workload characterization, which means developing a quantitative description of the workload, usually in terms of its impact on resource utilization.

Once the workload is characterized, each resource must be measured separately. For this task, a data collection and analysis tool, such as HP LaserRX, is essential to systematically and comprehensively record performance data. Raw or reduced data is used to analyze system activity, identify workload patterns, and measure and evaluate performance.

And just what is good performance? Ultimately, data processing departments and end users define performance standards between them. Ideally, this definition is formalized in a service level agreement.

User Service Level Expectations

Capacity planners must examine what level of service is provided to users and whether it meets their expectations. In general, user expectations are met when response times are acceptable or, for batch processing, turnaround times are short. Expectations regarding timeliness, accuracy, reliability, and cost should be quantified so that they can be measured and reported over time.

Business Plans

It is important to remember that capacity planning is part of the strategic planning of the company. More so than any other performance evaluation discipline, capacity planning is not a data processing function, but a business function. As a consequence, capacity planners must communicate with business planners in order to incorporate business factors into their analyses.

This means that capacity planners must have access to the strategic plans of the business. Is the company growing or shrinking? Is it profitable? Where does management plan to take the company? Do they plan to rely more or less on computers than they do now? What percentage of the budget will be allocated for hardware and software acquisition?

Furthermore, capacity planners should understand what their company's business units are and be able to relate these units to different workload metrics. Then they must get access to company business plans in order to predict resource utilization based on business units.

Predicting Workload and Business Changes

Workloads tend to grow, and as they do, system capacity becomes saturated and performance deteriorates. It is important to predict when the workload will exceed current capacity or performance limits in order to avoid both these outcomes. This means forecasting the growth rate of the existing workload, as well as the amount of new work to be added.

Typically, business plans and user feedback are prime sources for workload projections. User input is usually expressed in business, not data processing, terms. A capacity planner must be able to convert business data into meaningful units for forecasting purposes.

Workload forecasting includes visual trending and time series regression models.

A forecasting tool, like HP RXForecast, helps to predict system workload. HP RXForecast bases its forecasts on performance data collected by the HP LaserRX SCOPE collector.

HP RXForecast also accommodates using business plans to forecast. Its business unit trending method is used to relate such business units as customer phone orders to workload growth projections.

Setting Performance Objectives

The next task in the capacity planning process is to set performance objectives. Frequently, rules of thumb are relied upon to do this. For instance, that average CPU utilization should not exceed 70 percent. Obviously, using rules of thumb is relatively easy and low-cost.

It is not sufficient, however, to set performance objectives along a data processing bias. Fundamentally, computer systems exist to service users. Therefore, user expectations regarding response time, batch turnaround time, and system availability are essential when setting performance goals.

Of course, user service level expectations can be unrealistic. Data processing departments and users must agree to performance objectives that can be achieved within cost, quality, and technological constraints.

Many data processing (DP) organizations have service level agreements (SLA) with their users. SLAs are written agreements that set specific goals towards which the DP department works. They describe what and how much service is to be provided, to whom, and over what period of time.

In order to meet these goals, DP must have effective tools for monitoring system performance, as well as for problem diagnosis, application optimization, and capacity planning. This means a more proactive approach to system management that translates into greater economies of time and money, as well as increased user satisfaction.

It is usually necessary to translate user service-level objectives into system resource utilization thresholds. These thresholds are pre-determined values that, when exceeded, could lead to failure to meet user service-level expectations.

Thresholds are determined by monitoring performance data over a period of time for a particular system in order to deduce rules of thumb for that system. Say, for example, that it has been observed that response times become erratic when CPU utilization exceeds 70 percent. So a rule of thumb derived from this observation would be that average CPU utilization should not exceed 70 percent. Similar thresholds could be set for other system resources.

Performance objectives—stated in terms of either user service-level expectations or system resource utilization thresholds—can be compared to predicted performance in order to identify potential problems in time to avoid them.

Modeling to Predict Performance

The next step is to predict how the system will perform in the expected future. This is a major undertaking. Computer performance analysts use a number of models to predict future system performance.

A model is essentially an representation or description of some aspects of a system's behavior. Analysts use a model to predict what happens if certain actions are taken. They experiment with the model rather than use the real system. By so doing, they can answer "what if" questions about proposed alternative configurations without investing in expensive and potentially unnecessary hardware.

Figure 6185-1 shows the range of techniques that are used to model performance.

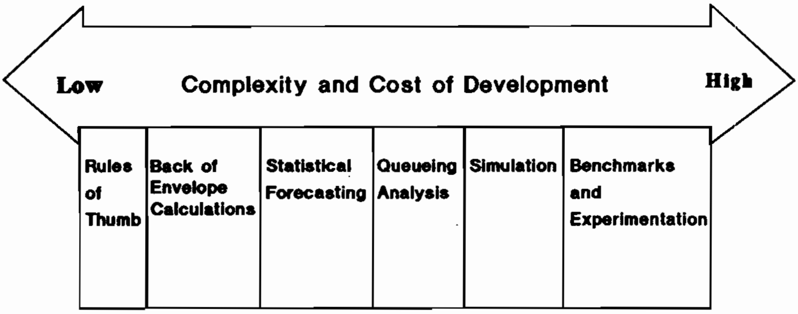


Figure 6185-1. Range of Computer Performance Modeling Techniques

The techniques increase in complexity *and* in cost of development from left to right. Rules of thumb is the simplest and least costly modeling technique, while benchmarking is the most costly and difficult to implement.

The accuracy of any technique's prediction, however, does not necessarily increase with either its cost or complexity.

Using rules of thumb or back-of-the-envelope calculations first is often the best approach because either one can eliminate using a more costly method unnecessarily.

Rules of Thumb

Rules of thumb are the least expensive and easiest to use of the techniques. They often represent collective wisdom: the knowledge accumulated by various system managers with various configurations.

For example, a common rule of thumb for interactive systems is that total CPU utilization should not exceed 70 percent for more than 10 minutes during any part of the day.

Rules of thumb can mislead, however, if they don't really apply to the system in question. The most useful are system specific: that is, they are developed over a long time from many observations of a particular system.

Even rules of thumb developed from experience with a system can become misleading if any of the components of that system, such as the operating system or CPU, are changed.

Some rules of thumb are not system-specific. For example, response times for a given application will fall within an expected distribution pattern regardless of the computer system upon which it is run. Just such a rule of thumb might be that 1 out of 10 transactions take 2.3 times as long as the average transaction; and that 1 out of 20 transactions take about 3 times as long. Suppose average response time for the application is 3 seconds. If the system is behaving normally, the expectation would be that 1 out of 20 transactions will be less than or equal to 9 seconds.

Back-of-the-Envelope Calculations

Back-of-the-envelope modeling involves informal calculations, as the name implies. Analysts use this technique to make rough estimates about a proposed action's feasibility.

For example, suppose 20 additional interactive users are to be added to a system. Also the application they use is CPU-bound; that is, in this environment, CPU would be the first physical resource (as opposed to I/O) to bottleneck.

Each of the 20 users adds a load on the CPU of about 1.5 percent. If current average CPU utilization during the busiest period of the day is 70 percent, adding 20 users could bring CPU utilization up to 100 percent and cause serious performance problems. An alternative scenario, such as rescheduling some of the current workload or upgrading the computer system, must be planned.

Back-of-the-envelope modeling is especially useful for making quick estimates of the effects of adding new workloads to a current system or upgrading to a larger system. It is sometimes used with other more advanced modeling techniques as a reasonableness check.

Statistical Forecasting

Many forms of statistical forecasting exist, but all of them use collected performance information (logfiles) to establish a trend, which is then projected into the future to forecast performance data. Simply put, the future is extrapolated from the past.

Statistical forecasting techniques are more precise than either rules of thumb or back-of-the-envelope calculations because they use actual data on a specific system to make projections.

Statistical forecasting techniques rely on such methods as exponential smoothing and regression analysis. Consequently, capacity planners benefit from a forecasting application like HP RXForecast, which uses time series analysis and regression analysis.

HP RXForecast uses data collected and reduced by HP LaserRX to display trends and make forecasts. One big advantage of HP RXForecast is that it relates user-defined business units to measured workload metrics in order to model.

Queueing Theory

Computer analysts have adopted queueing theory (or the theory of waiting lines) for modeling computer systems.

Queueing models are sets of mathematical formulas that are constructed to capture the relationships between workloads and performance measures. A basic queueing model of a computer system consists of a source of potential customers (incoming work or transactions), at least one waiting line or queue, and at least one service center (a CPU, I/O subsystem, etc.).

A queueing theory formula is:

$$R = \frac{D}{(1 - U)}$$

Where:

R is the average time for a request to be serviced; that is, response time.

D is the average demand.

U is the average utilization. That is, the fraction of the time that the server is busy.

If the server is busy 80 percent of the time, U = 0.8.

$$R = \frac{D}{(1 - 0.8)}$$

Then R = 5D.

This means that 4D units of time are spent queueing for service to begin and 1D is spent in actual service.

On the other hand, if the facility is busy only 60 percent of the time, U = 0.6.

$$R = \frac{D}{(1 - 0.6)}$$

Then R = 2.5D.

This means that 1.5D units of time are spent queueing for service to begin and 1S is spent in actual service. Lowering utilization from 80 to 60 percent significantly lowers queueing time.

Figure 6185-2 is a graphic representation of the queueing theory formula above.

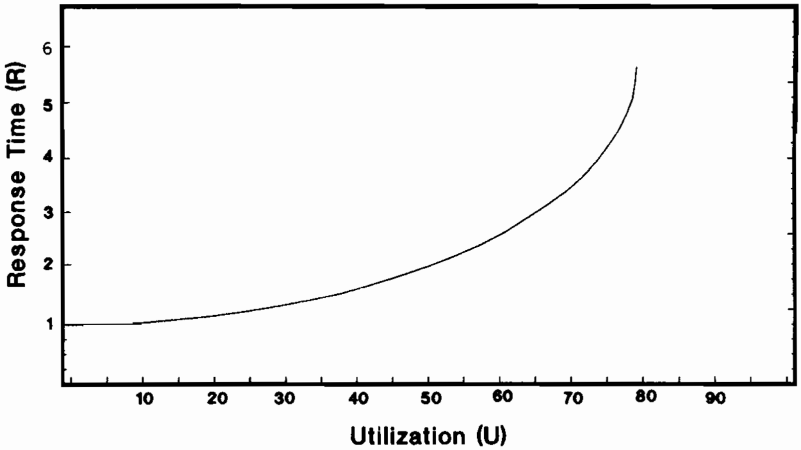


Figure 6185-2.

For lightly loaded systems, response time is simply the service time of the facility—remember when you were the *only* one on the system? For heavily loaded systems, waiting or queueing up at the facility begins above 80 percent utilization.

This formula explains many of the delays we experience every day and why service deteriorates as computer systems become more heavily loaded.

Simulation Modeling

Simulation modeling is a an experimental method that mimics the system being modeled. That is, it is a method in which a system is analyzed by observing how a model of it responds to artificially generated input.

In simulation modeling, the following procedures are followed:

- Transactions are constructed and routed through the simulation model just as they would flow through the real system.
- Simulation modeling software keeps track of how long each transaction spends in each service center of the model.
- From these counts, it constructs the performance statistics.

  In order to get reasonably precise performance statistics, a large number of transactions must be run through the model.

Queueing Theory and Simulation Modeling Compared

Either a queueing or simulation model can be modified to answer such hypothetical questions as the following:

- If I add 40 new users to my interactive system, what will the new average response time be?

- If I procure new disk drives that are twice as fast as the old ones, what will be the impact on performance?
- How many interactive users can be supported if CPU capacity is doubled?

A queueing theory model can be solved using either a simulation or an analytic approach. In both cases, the model is an abstraction of the system (real world); the solution is an approximation based on a number of assumptions.

Queueing theory modeling and simulation modeling are the two types of advanced modeling techniques most often used by capacity planners.

Queueing theory models are easier to construct, easier to validate, and require less computer time to run than do simulation models.

However, simulation models can be used to model systems in more detail than queueing theory models and can model some systems that are beyond the reach of queueing models.

Benchmarking

Benchmarking is the process of selecting a representative set of programs, running them on the existing system to establish a "benchmark" time, and then running them on other systems to compare performances. They are most appropriate for comparing a new system to a current one for an existing workload.

To be representative, a set of programs must, when run as a mix, produce a load on system resources that is proportional to the load imposed by the entire workload. That is, the benchmark measures will be approximately the same as those derived from the full workload during peak processing times. Peak processing times are important because performance analysts are generally interested in how a system performs under heavy workloads.

Full-scale benchmarking can be very expensive and time-consuming. It requires some specialized expertise to construct and run.

Some installations develop an in-house benchmark to evaluate system changes. They then have a readily available measurement tool that becomes increasingly cost effective with each use. Such a tool can also be used to track system performance over time.

Validating Models

Any model must, of course, be validated before its performance predictions can be considered reliable.

Validating a model means confirming that it reasonably represents the system that it is designed to represent. The validation procedure consists of experimenting with the model using measured data.

One way of validating a model is to use actual, measured parameters from the computer system to set up the model and then, after running the model, to compare performance metrics from the model to those from the measured system. If the results are reasonably close, the model is considered valid.

Common sense is easily the single most important tool for testing a forecast's validity. Statistical confidence tests can also be used to determine the significance of any differences between calculated and measured data.

Computer Museum

Forecasting packages like HP RXForecast have a validation feature that allows the user to test a forecasting method. The initial part of actual, recorded performance data is compared to predicted data to evaluate the model.

Evaluating Alternatives

A capacity planner must also predict how alternative configurations would perform. It may be necessary to plan some orderly upgrades to meet projected requirements, whether by adding more memory or faster I/O devices, by splitting up the workload, by adding a more powerful computer, etc.

Maintaining Capacity Planning Information

In this paper we have tended to look at capacity planning as a series of snapshots. The capacity planning process starts with a snapshot of today: the system is measured and the workload, business environment, and user service level are analyzed. Then another snapshot is taken, this time of the future: where is the business going, what will the workload be, what will users expect, etc. Then a hypothetical snapshot of expected future system performance is taken. Alternatives are weighed from data processing and business points of view. And appropriate changes are made to the system configuration.

But capacity planning more closely resembles a moving picture. The future rapidly becomes the present, and complaisant capacity planners find themselves in a purely reactive mode again, wondering why users are complaining so much about response times.

The effective capacity planner maintains system performance data on a long-term basis; ideally, indefinitely.

Actual business and workload growth should be reconciled against growth projections, and actual system performance should be compared against predicted—or modeled—performance.

Frequent snapshots of today must be taken: users must be talked to often, the workload must be analyzed as it changes, tracking and reporting mechanisms must be instituted, and changes in the business environment must be monitored.

C++: A New Software Development Methodology

Michael Beckmann and Dmitry Lenkov

Hewlett-Packard Company
California Language Laboratory
19447 Pruneridge Avenue, MS: 47LE
Cupertino, CA 95014

1. Introduction

Since the development of the first programs, software developers have had to deal with applications which are unreliable, inflexible, and not easily modified. These shortcomings can be attributed in part to the ever-growing complexity of the problems for which software developers have set out to provide solutions.

In the early days of computing, software development was extremely tedious. Detailed knowledge of the underlying hardware was required to develop software for these early systems. The programs were written in numerical sequences and later mnemonic encodings, known as machine or assembly code. It became apparent in the 1950's that this form of programming was becoming unmanageable and limited programmers' ability to develop and maintain reliable software. This brought on the development of the first high-level programming languages, namely FORTRAN and COBOL. These early languages were extremely successful in providing solutions to the problems they set out to solve at that time. Unfortunately, since the creation of FORTRAN and COBOL, and more recently languages such as C and Pascal, software systems have been steadily growing in complexity to the point that these languages no longer adequately address the problems associated with large software systems.

In an effort to curb the problem of growing complexity, software developers have turned to the new methods of object-oriented programming languages. One such language is C++. C++ is a general purpose, C-based programming language which supports object-oriented programming. C++ is now widely perceived as one of the best suited languages for software development. While the popularity of C++ is enthusiastically supported by the C community, the main reason for such growth is that companies are finding C++ of strategic benefit for their development of software.

In this paper we analyze some of the problems associated with software development and the advantages to be gained through the use of object-oriented programming with C++. We also analyze the importance of the standardization of C++ and describe Hewlett-Packard's expertise in the area of object-oriented technology.

2. Problems Associated with Software Development

To better understand the problems associated with the development of large software systems we will examine the notion of complexity in software and some of the desirable qualities of software which are often compromised [1] [2] [3].

2.1. Complexity

It has been said that good programming requires the mastery of complexity. The growing complexity of software systems is the single most important problem developers need to overcome. Software complexity is an underlying problem which leads to other undesirable software traits such as reliability and maintenance problems. Complexity in software exists because of the growing complexity of the problem domains which developers are trying to address. The complexity of the problem domain is usually magnified when abstracted through software. This additional complexity is incurred in part by the design methodologies chosen.

An important aspect of software methods is the programming paradigms employed. Programming paradigms can be thought of as a patterned approach to problem solving influenced and limited by the linguistic feature sets provided by programming languages. FORTRAN is considered to be a procedural language which imposes a procedural or structured programming paradigm. Problem solving using the procedural programming paradigm leads to functional decomposition of a problem until the problem becomes manageable. Programming paradigms strongly influence design methods, and should therefore be thought of as included in software methodologies.

The influence of programming paradigms on software design is apparent in various design techniques. Traditionally, structured and data-structure design methods have dominated the mind-set of software developers. This is reflected in the feature sets of today's conventional programming languages. With structured design, the focus is on the systematic functional decomposition. Structured design, however, has some drawbacks: with functional decomposition the emphasis is on the operations needed to solve a problem as opposed to the data associated with a problem. Data-structure design techniques focus on just the opposite. With this design method, the emphasis is on creating the data representation of a problem and then designing the complete system around the data. Both structured and data-structure design methods suffer from inter-dependencies between functional units. Ultimately, these traditional design techniques have not done much to help with the management of complexity.

2.2. Reliability

Software reliability is the ability of software to perform its designed tasks without failure in its operation. Software reliability is clearly one of the most important aspects of software development. Without reliability no great measure of assurance can be made against catastrophic failure of computer systems. Reliability is greatly influenced by the design methodologies used in the development of software applications. The design of a software component so that it incorporates modularity and localization, as well as understandability, is paramount to the reliability of a software system. Over the life-cycle of an application reliability is often compromised due to bad design choices in early development of that application. Because of the implications of design methods on reliability, special attention should be given to programming paradigms.

2.3. Maintainability

Maintainability of software is the ease at which software can be modified because of either changes in the design requirements, efficiency improvements, defect repair, or a change in hardware. The maintenance of software involves changes over the life-cycle of an application. Maintainability is an important software quality because of the large percentage of cost spent in the maintenance of a software system. It is estimated that 70% of the cost of software is attributed to maintenance [4].

The modifiability and understandability of software directly reflects its maintainability. As with reliability, design methods play an important role in the maintenance of software.

2.4. Reusability

Reusability is the ability to reuse existing software components for new applications. Reusability is important to software development because of the cost involved in developing new software systems. In the development of software, particular implementations and solutions to problems repeat themselves. Fundamental data structures are continualy being redesigned in software systems. A great deal of time is spent redesigning solutions which are well understood. Linked list manipulation and sorting solutions are just a few of the applications of this kind. The reusability of software components requires flexibility and the ability to extend existing components without changing them.

3. Object-Oriented Programming

Software development can be viewed as systematic methods of managing complexity. When designing software applications we typically represent or model real world problems through abstractions in software. Object-oriented design is a method which models real world problems with objects. So what is "object-oriented programming"? Object-oriented programming is a programming paradigm where problems which exist in the physical world, either real or conceptual, are modeled through the manipulation of objects. Object-oriented programming is based on a few key concepts which embody the object-oriented programming paradigm [5] [6].

3.1. Objects and Data abstraction

The notion of an object is central to understanding the concept of object-oriented programming. Objects are software abstractions of real world entities represented as data structures and operations on the data structures. They are organized as independent entities which have values or states. Objects support data abstraction, a technique for defining a new data type, or a class of objects, in terms of operations on that data type. Data abstraction supports the manipulation of objects through an operational interface which allows hiding of the data formats and algorithms. Such hiding of implementation details provides for reducing complexity.

3.2. Encapsulation

Encapsulation is the concept of hiding implementation details of an abstract data type. This allows for the separation of the specification and the implementation of objects or classes of objects. Keeping details of the implementation hidden contributes to reducing the complexity of software systems. It is also important from a software modifiability and maintenance viewpoint. The interface of an object does not require changes when the underlying implementation changes. Therefore, code accessing objects does not require a corresponding change. Encapsulation enforces independence of software abstractions.

3.3. Inheritance

Inheritance is the notion of being able to define an object in terms of other objects. Inheritance allows object definitions, both data and operations on the data, to be shared so that new objects can be created incrementally by the extension of existing objects. This also allows for common abstractions to be represented uniquely. Inheritance, in general, has two main advantages. It allows reuse of existing object interfaces and implementations without code replication. It also supports *polymorphism*, since operations defined for a class of objects can be applied to objects of all classes in the hierarchy of object classes inheriting from that class.

3.4. Polymorphism

Polymorphism is the ability to manipulate different types of objects, which share a common interface, by issuing generic calls to functions or operations on objects. This is useful because it focuses on the operational interface of an object rather than the actual object. This concept further facilitates code reuse by allowing common interfaces on many different types of objects. Polymorphism supports extendibility of objects through increased generality.

4. Advantages of Object-Oriented Programming

In this section the benefits of object-oriented programming are presented. Object-oriented programming techniques, namely, the use of data abstraction, encapsulation, inheritance, and polymorphism, offer many advantages over conventional programming methods.

4.1. Ease of Modeling

The object-oriented methodology of software design addresses the problems associated with traditional design techniques by allowing the designer to more closely model the world through the manipulation of objects. The world as we perceive it is made up of objects, either real or conceptual. For this reason, a programming paradigm that allows a designer to pattern software more closely to these real world objects has clear advantages over other paradigms. A closer modeling reduces the added complexity gained through the software abstraction of the problem domain. Ultimately, the reduction in complexity benefits overall software reliability and increases understandability.

4.2. Code Reusability

The ability to reuse existing software is possibly one of the greatest benefits to be realized from object-oriented programming techniques. With code reuse, the cost of developing and maintaining software is significantly reduced. Reusability of software is provided through the effective use of *inheritance* and *polymorphism*. Inheritance supports incremental development which is extremely important for increased productivity when designing new applications as well as extending old applications. Inheritance is an important concept because it increases the flexibility of the designer to create new applications by the extension or restriction of existing code. Also, inheritance provides a mechanism to avoid repetition of software design. The concept of polymorphism is a natural extension of inheritance. Polymorphism allows software to be extended without changes to existing software. The exact type of the object is not as important as the interface. If a consistent interface among objects of a common nature exists, then, through the use of polymorphism, the objects can be manipulated in a generic fashion.

4.3. Reduced Maintenance Costs

The concepts of *data abstraction* and *encapsulation* provide an excellent mechanism to isolate implementation details such as algorithms and data formats of objects. By clearly separating the interface from the implementation, certain parts of the software can be changed without altering or affecting other parts of the software. Software components can now have well understood and well defined functional interfaces which help increase reliability and reduce maintenance costs. Reduced maintenance costs can be also be realized through code reuse. Reused software components usually benefit from increased reliability.

5. C++ as a new Software Development Methodology

C++ is a general purpose programming language, designed by Bjarne Stroustrup of AT&T Bell Laboratories. C++ was designed to be a better C by supporting data abstraction and object-oriented programming, while retaining the compatibility and efficiency of existing C programs. Developed in the early 1980's, the evolution of C++ can be traced back to Simula67, Algol68, and of course C, hence the name C++ ("++" is used as the increment operator in C) [5] [7] [8] [9] [10].

Why use C++? It is widely accepted that code reuse and increased manageability and reliability of developed software are major advantages for moving to C++. In addition, C++ is currently the fastest growing object-oriented language on the market. One particularly attractive attribute of the language is the transition path it provides from C. For programmers familiar with C the use of C++ can be considered more as an extension to C. Other object-oriented languages tend to have a longer learning curve. The efficiency of C++ is comparable to that of C, an advantage over other object-oriented programming languages.

5.1. C++ Support of Object-Oriented Programming

C++ provides various linguistic features which support object-oriented programming. Some of these features will be presented in relation to the object-oriented concepts previously described in section 3.

5.1.1. Data Abstraction

C++ provides support for data abstraction. With C++ a user can define types that behave much like built-in types. The concept of data abstraction is facilitated through the linguistic feature of a *class declaration*. The declaration of a class in C++ specifies the representation of data and operations on that data. The operations are made available in the form of functions which are part of the class. These functions are typically referred to as *member functions*. Member functions can only be invoked for a specific object or instance of that class. Below we will present a popular example used in C++ [5]:

```
class complex {
private:
      double real, imag;
public:
      complex(double r, double i) {real = r; imag = i;}
      complex(double r) {real = r; imag = 0.0;}

      friend complex operator+(complex, complex);
      friend complex operator-(complex, complex);
      .
      .
};
```

In this example we have defined a class which has two *data members*, **real** and **imag**, representing the real and imaginary part of a complex number. Along with the data members, we have also provided operations on the data in the form of member functions. Special member functions have been provided to initialize an object of type **complex**. Also, the "+" (addition) and "-" (subtraction) operators have been provided to manipulate **complex** objects.

Included in this example is the use of a C++ feature known as *overloading*. Both functions and operators can be overloaded. In this example the operators "+" and "-", which are defined for the C built-in types such as **int** and **float**, are overloaded so that the addition and subtraction of two **complex** objects can be performed. Another use of overloading is shown through the two definitions of the function **complex**. This is an example of function overloading. Function and operator overloading are very important features needed to support data abstraction and polymorphism because they allow for transparent access and increased flexibility of user defined types. Overloading is provided in C++ through the type checking mechanism. C++ disambiguates operators and functions by the type and number of arguments associated with them. Class association is also used.

Another important feature of C++ is shown in this example, namely *inlined functions*. Both of the definitions for the function **complex** are inlined. A function is referred to as inline when its code-body can be expanded where it is referenced. By inlining a function the overhead of the function call is avoided. Functions are specified inline as a result of defining the function bodies in the class definition or by specifically declaring them **inline**. Inlined functions will be discussed in detail in section 5.2.2.

Now that we have presented the declaration of the type **complex**, we will show how objects of that type behave like built-in types. Uses of the addition and subtraction operators are included in this example. By expanding the definition of these operators, through the mechanism of overloading, objects of the type **complex** can be used transparently, independent of their underlying implementations:

```
main() {
    complex x(4.5,2.5), y(1.0), z(0.0);    // definition of x, y, z
    z = x + y - z;
    .
    .
}
```

In this example we have defined three objects of type **complex**. The object **x** has **real** and **imag** values of 4.5 and 2.5 respectively, while the object **y** has values 1.0 and 0.0. The object **z** has both values set to 0.0. In the definition of both **y** and **z**, the overloaded function **complex**, with a single argument of type **double**, is called. Both of the overloaded functions **complex** are referred to as *constructors*. Constructors will be explained in greater detail in the section on encapsulation.

We will now present another declaration of a class in order to illustrate other features of C++ that support object-oriented programming:

```
class account {
private:
    char *name;
protected:
    double balance;
public:
    account(char *new_name);

    void deposit(double amount) {balance += amount;}
    void withdraw(double amount) {balance -= amount;}
    virtual void update_balance() {}
    virtual void print_statement() {cout << name << " " << balance << "\n";}
};
```

In this example we have defined the class **account** which is a simple representation of a bank account. An object of type **account** would contain a **name** and the balance of the account. Also, member functions are provided to support operations which are associated with an account. This example shows the use of *access specifiers* which are a feature of C++ used to support encapsulation.

5.1.2. Encapsulation

Support of encapsulation is provided in C++ through the use of access specifiers. Referring to our previous example defining the class **account**, the class declaration contains access specifiers which control the interface and internal visibility of an object of that class. In C++ this is done with the **private, public,** and **protected** keywords. If a data member or member function is specified as **private** it can not be accessed outside of that class. Only member functions can have access to a private member. If a member is specified as public it is visible and can be accessed anywhere within its scope. The **protected** specifier is like **private** access except that it allows *derived classes* to have access to members of the *base class*. Derived and base classes will be presented in the section on inheritance.

Another form of encapsulation can be found in the separation of the class specification and its definition. As in the **account** example, a public member function is declared; however, its definition exists in another module. This is very useful in hiding the implementation of classes without limiting their use. As an example we will now define the function **account** which creates an instantiation of the class **account**:

```
account::account(char *new_name) {
      name = new char[strlen(new_name)+1]; //create a new string
      strcpy(name, new_name);              //copy new_name to safe string
      balance = 0.0;                       //initialize balance
}
```

To adequately support data abstraction and encapsulation, a mechanism needs to be provided to initialize objects in a manner such that the representation of the object can be hidden. To accommodate this requirement C++ provides for the specification of member functions to initialize and clean up objects of a given type. These special member functions are referred to as *constructors* and *destructors*. When defining an object of a given type, a constructor is called at the creation of the object and the destructor is called when the object is to be destroyed. The function **account** is a constructor for an object of the class **account**.

5.1.3. Inheritance

C++ supports the concept of inheritance with the feature of *derived classes*. In C++, a class can easily be defined as parts of existing classes referred to as *base classes*. A class may be declared as a derived class of a base class by specifying the base class on the class derivation list. We will now define two classes which are derived classes from the base class **account**:

```
class checking_account : public account {
private:
      double fee;
public:
      checking_account(char *new_name) : (name) {fee = 2.00;}
      void update_balance() {balance += (balance - fee);}
      void print_statement() {
            cout << "checking\n";
            account::print_statement();
      }
};
```

```
class savings_account : public account {
private:
      double rate;
public:
      checking_account(char *new_name) : (name) {fee = .05;}
      void update_balance() {balance += (balance * rate);}
      void print_statement() {
            cout << "savings\n";
            account::print_statement();
      }
};
```

In this example we have created two derived classes which inherit the base class **account**. All of the member functions provided in the base class can be used on objects of the derived classes. The function **deposit** can be used to deposit money into a **checking_account** or **savings_account**. The function **print_statement** in either derived class calls **print_statement** from the base class through the use of the "::" (scope operator). The feature of derived classes is a very powerful mechanism supporting code reuse.

5.1.4. Multiple Inheritance

The examples presented up to this point have only shown the use of single inheritance. Single inheritance means that a class is derived from only one base class. With multiple inheritance a class can be derived from more than one base class. Multiple inheritance is useful because an object can be made up of many sub-objects. C++ supports the mechanism of deriving a class from multiple classes. Creating a class which is derived from multiple classes is done by specifying the base classes on the class derivation list. For example, the class **savings_account** could have been derived from the class **account** and possibly a class **investments** which represents stock holdings, mutual funds, etc:

```
class savings_account : public account, public investments {
      .
      .
};
```

5.1.5. Virtual Functions and Dynamic Binding

To support polymorphism, C++ provides *virtual functions*. A virtual function is an overloaded member function which is specified in the base class. Any class which is derived from the base class can redefine the virtual function. By specifying a function as virtual, C++ can perform *dynamic binding* of member functions, based on the type of the object being referenced. In the **checking_account** example, the member functions **update_balance** and **print_statement** are declared as virtual. This allows the following to occur:

```
account *account_ptr = new checking_account("Joe Smith");
account_ptr->update_balance();
account_ptr->print_statement();
```

In this example we have defined a pointer to the base class **account**; however, we initialize it so that it addresses an object of type **checking_account**. At run-time, **account_ptr** can address either a derived or base class object. By declaring **update_balance** and **print_statement** as virtual functions, C++ will select the proper function, based on type, to be executed.

5.2. Other Features of C++

C++ provides other useful features which make it a superior general purpose programming language. These features do not directly support object-oriented programming; however, they have been included because of the increased reliability and flexibility which they provide [5] [7] [8] [9] [10].

5.2.1. Type Checking and Coercion

Type checking and coercion do not directly support object-oriented programming; however, other features such as function and operator overloading, derived types, and virtual functions could not be provided without a strong typing mechanism. One common reliability pitfall which occurs in some programming languages is that no assumptions are made about the type and number of arguments of a function. This allows a function to be called with an unexpected number and/or type of arguments which could lead to an error when the program is executed or, worse, an invalid result could be returned and go unnoticed. C++ has remedied this problem with the introduction of *function prototypes*. With function prototypes, a function and its argument list are declared explicitly, specifying the number and type of the arguments as well as the function's return type. Examples of function prototypes are:

```
int   max(int i, int j);
void  push(Stack_Object obj);
```

Function prototypes have proven to be so valuable that ANSI has adopted them in the new standard for the C programming language.

5.2.2. Inline Functions

Inline functions are another useful feature of C++. The use of inline functions makes it easier to localize changes to an occurrence of a common computation. Also, the functional interface to class objects is preserved without the overhead of a function call. The main motivation for inline functions comes from the use of macro definitions in C. Inline functions protect against the use of incorrect types which can occur in macro definitions. An example of an inline function declaration:

```
inline int max(int i, int j) { return ( i > j ? i : j ); }
```

5.2.3. Scoped and Typed Constants

The introduction of symbolic constants allows for the use of objects whose value can never change. A symbolic constant can be viewed as a read-only data type which may not appear on the left hand side of an expression. This is a big advantage over traditional C and the C preprocessor in that symbolic constants can be type-checked as well as scoped. To define a symbolic constant the **const** specifier is used:

```
const int max_str_size = 1024;
const int *max_ptr = &max_str_size; // max_ptr is not a constant
```

In this example, a symbolic constant of type **int** is defined to have the value of 1024. In defining **max_ptr** we have created a pointer that addresses a constant. **max_ptr** itself is not a constant and can be changed to point at another constant object.

5.2.4. Argument Passing

C++ provides a number of useful features in the area of argument passing. It is common in C to call functions with a varying number of arguments and to pass arguments of different types. C++ provides a mechanism in which a formal parameter can be assigned a default value. As an example:

```
array resize_array(array *ar, int new_size = 256);
```

Another useful feature is the ability to suspend type checking when the number and types of the arguments are not known. Type checking can be suspended with the specification of "..." an (ellipsis) in the formal argument declaration. The function **printf** is declared with an ellipsis:

```
extern int printf(const char *, ...);
```

Sometimes it is useful to be able specify the calling mechanism to be used in argument passing. C++ allows the use of reference declarations. *call-by-reference* is helpful when passing large objects or when an argument is to be modified. As an example of a reference declaration we will define a function that exchanges **num1** with **num2:**

```
void exchange(int &num1, int &num2);
```

6. C++ Standardization and its Importance

The requirements for the portability of C++ programs, for a full specification of the C++ language, and for a specification of C++ libraries and supporting environment have reached a level of importance similar to that of other widely used and mature programming languages such as C and Pascal. Standardization of the language is a very important factor in satisfying these requirements [11] [12].

6.1. Portability

Portability is one of the main requirements of the modern software development process. The portability of C programs is one of the major reasons for the success of C. Programming language portability is based on an accepted language definition, consistency and completeness of language features, and a standard set of library functions, as well as unambiguous identification of portable and non-portable features.

The number of different C++ implementations is growing rapidly. In the absence of a full language definition, many implementations introduce their own interpretation of some C++ features. This leads to various language dialects. Multiple dialects of a language damage its portability and introduce serious problems to both suppliers of language products and their users. Experience with such languages as FORTRAN and Pascal clearly demonstrates that.

6.2. Compatibility

Compatibility is closely associated with portability of a language. There are two different levels of compatibility.

Source level compatibility is defined as the ability of different language processors, for a particular language, to treat the same program in this language in a semantically equivalent way. A standard definition of such languages assures source level compatibility.

Run-time compatibility allows outputs (modules) produced by different language processors for a particular language to be linked and executed together. This kind of compatibility is very important for object-oriented languages in general and for C++ in particular. Reuse of existing code such as C++ class libraries is one of the major factors in this. A standard definition of a language normally provides guidelines for achieving run-time compatibility.

6.3. Missing Language Features

The current implementation of C++ by AT&T and the available language definition of C++ do not provide certain important features that are needed for large and medium size development projects. Among them are exception handling and support of generic functions and types. Because of this there are already implementations that support these features, or some of them, in various ways. Such implementations are now available within certain organizations internally. However, implementations that are commercially available can be expected in the near future, as well as a significant increase of their number. This expands the danger of multiple dialects which leads to non-portable programs.

6.4. Libraries

For many languages, a standard set of library functions has become a part of the language definition. This is obviously true for C. Such a set of library functions provides an extension of language functionality without overloading the language with features. At the same time, it satisfies the needs of software development in some particular areas not covered by the language features.

The functionality of C++ libraries, as well as a standard set of header files, is not completely defined. This causes almost every development project using C++ to develop their own additional libraries.

6.5. Supporting Environment

The role of a language supporting environment is much greater for object-oriented languages than for languages like C and Pascal. Smalltalk and Ada environments clearly demonstrate this. A C++ supporting environment may be the only reasonable way to allow more independence of class interfaces from class implementations, facilitate header file management, and solve some other language related problems. Several existing C++ supporting environments are completely different. Further development in this direction will create semantic differences in the interpretation of some language features.

6.6. Fear of a Language with No Standard Definition

Many development projects find C++ to be the most appropriate language for their needs. However a large number of them do not choose C++ because of the instability of the language definition and the absence of any effort toward its standardization. Alternatively, they choose C, Pascal, or some other language of this level and implement their own extensions which allow the support of object-oriented programming with the chosen language. This definitely increases problems with portability and code exchange.

6.7. ANSI C++ Committee (X3J16)

In September of 1989 the ANSI X3 Secretariat approved the proposal for the standardization of the C++ programming language, submitted by Hewlett-Packard, and formed a new technical committee (X3J16) for this purpose. X3J16 had its organizational meeting on December 15, 1989, in Washington, D.C. The meeting brought together 49 C++ experts and observers representing 40 organizations and individuals. Discussions primarily focused on the committee goals, scope of work, and of planning future meetings.

The first technical meeting was held March 12-16, 1990, in New Jersey. Over 50 people participated in this meeting. Main discussion topics included: early international participation, scope of work, and Bjarne Stroustrup's C++ reference manual. Both meetings were viewed as a successful beginning of the C++ standardization process. The scope of work accepted by the committee includes the standardization of currently available features of C++, libraries, and C++ environment, as well as consideration of major proposed extensions to the language. It is expected that the committee will be able to produce a solid draft document by the end of 1992.

One of the important results of these two meetings is that the committee members found an early international participation and close collaboration with international and other national standard organizations on the production of a single draft standard.

7. HP's Expertise in Object-Oriented Technology

Hewlett-Packard is among the leaders as a supplier of products supporting object-oriented programming as well as a user itself of object-oriented technology. HP's expertise includes research in the areas of conceptual and theoretical models of object-oriented systems, object-oriented design techniques, development of products supporting object-oriented programming, and use of object-oriented languages and systems for software development.

7.1. Research and Investigations

HP supports and initiates various research and development projects in the area of object-oriented technology. The results have been reported in various conferences and in magazines and journals. For example, at OOPSLA'89 (conference on Object-Oriented Programming Systems, Languages, and Applications), HP scientists and engineers delivered 5 papers, and participated in panels, workshops, and various bird-of-feather sessions.

Among the projects in the area of object-oriented technology, particularly noticeable was an effort in creating a glossary of common object-oriented terminology and conceptual object model. The common object model developed helped to better understand different object-oriented products and systems developed by HP. The glossary and the object model also generated a significant interest outside of HP. As a result they were adopted with certain extensions and modifications by OMG (Object Management Group).

7.2. New Wave

The New Wave product is an object-oriented environment for application integration. It provides an architecture for end user environments in which applications are easy to learn and use, and are cooperative. New Wave minimizes the development costs of supplying application integration services including cooperation, task automation, computer based training and context-sensitive help.

The major elements of the environment are organized following an object-oriented design. The conceptual model of the user's view of computation follows an object-oriented paradigm. It includes: graphical environment based on real world metaphors, objects representing the user's work products, and encapsulation of programs as objects with uniform interface.

7.3. SoftBench

SoftBench is a tool integration platform that provides improved tool communications and integration in a distributed computing environment. The foundation for tool integration in SoftBench is the Broadcast Message Server. Each tool in SoftBench sends out messages in well known formats describing what it has done, or requesting services from other tools. The Encapsulator product is provided to help migrate existing tools to the SoftBench platform. It allows the creation of a windowing interface to a tool integrated into SoftBench environment. It also allows existing tools to utilize the SoftBench message passing facility to interact with other tools in the environment.

7.4. HP C++

HP has provided C++ for HP9000 series 300, 600, and 800 systems. The HP C++ product supports the AT&T industry standard C++ language feature set and includes a symbolic debugger, libraries, support from HP-UX development tools, and extensive documentation.

The HP C++ symbolic debugger provides full C++ support for debugging object-oriented programs. HP C++ symbolic debugger features include class level and object level breakpoints and views, the proper handling of overloaded functions and operators, and supports correct handling of object type information (self-describing objects).

7.5. Use of Object-oriented Technology

Object-oriented technology has been used in HP for software development since the early 1980's. The number of projects that have used or are using this technology can be measured in at least dozens and may be well over a hundred. Currently, C++ is playing a larger role within HP. Multiple projects are either already using C++ for software development or in a process of switching to it. Particularly interesting is the case of one project at HP Clinical Information Systems Operation. It has developed a clinical information management system in C++. The object-oriented approach was believed as the best one providing techniques necessary to lead the project to a success. The amount of C++ code implementing that system numbers over several hundred thousand lines.

8. Conclusions

The mounting challenges of increased reliability, reduced maintenance costs, and increased productivity, along with the promise of software reusability, have caused developers of large software systems to turn to object-oriented technologies. C++ is one of the best suited general purpose programming languages available for object-oriented software development. C++ brings with it the power of object-oriented methods while retaining the advantages of C, such as portability, flexibility, and efficiency. Along with C++, Hewlett-Packard is involved in many object-oriented technologies in an effort to provide solutions that benefit software developers.

9. References

[1] Grady Booch, Software Engineering With Ada, Benjamin/Cummings, 1986. Chapter 2, "The Software Crisis". Chapter 4, "Software Engineering".

[2] Bertrand Meyer, Object-oriented Software Construction, Prentice Hall, 1988. Chapter 1, "Aspects of software quality".

[3] Peter Wegner, Object-oriented Software Engineering, Tutorial Notes, International Conference on Computer Languages, 1988.

[4] Bertrand Meyer, Object-oriented Software Construction, Prentice Hall, 1988. Chapter 1, Section 1.3, "About Software Maintenance".

[5] Bjarne Stroustrup, What is "Object-Oriented Programming?", USENIX Papers, 1987.

[6] Bertrand Meyer, Object-oriented Software Construction, Prentice Hall, 1988. Chapter 4, "The Road to Object-orientedness".

[7] Bjarne Stroustrup, The C++ Programming Language, Addison Wesely, 1986.

[8] Bjarne Stroustrup, The Evolution of C++: 1985 to 1987, USENIX Papers, 1987.

[9] Stanley B. Lippman, A C++ Primer, Addison Wesley, 1989.

[10] Ira Pohl, C++ for C Programmers, Benjamin/Cummings, 1989.

[11] Dmitry Lenkov, X3/89-738R, Programming Language C++ Proposal to ANSI X3 Secretariat. 1989.

[12] ANSI X3J16, Document X3J16/90-0023, 1990.

Optical Disk Technology: Past, Present, and Future

Husni Sayed
Deborah Cobb
Andrew Patterson
IEM, Inc.
P.O. Box 8915
Fort Collins, CO 80525

Introduction

Optical Disk Technology has become the latest "craze" in the computer industry. Though still in its infancy, this technology is quickly increasing in popularity as more users discover what it has to offer: namely, random access to a remarkable amount of data stored on a compact, removable cartridge.

Optical Disk Drives use lasers to store and retrieve data from optical disks. The CD-ROM (Compact Disk Read Only Memory) and WORM (Write Once, Read Many) technologies are currently available from a number of sources. Erasable optical disks drives, much newer to the field, have become commercially available from a few companies only within the past few months.

Optical Recording Methods

Optical disk technology was "fathered" by Phillips Corporation. CD-ROM and WORM drives are now being developed by such companies as Sony, Toshiba and Ricoh, along with a host of smaller companies. Erasable disks are being developed by Sony, Ricoh, and Hitachi (to name a few).

CD-ROM Technology

CD-ROM (Compact Disk Read Only Memory) disks are not generally useful for most mass storage applications, since they are "Read Only" memories: information can be written to the disk only during the manufacturing process, not by an end-user. With CD-ROMs, information is stored on only a single side of the disk. The surface of a CD-ROM disk has a continuous spiral track, like a phonograph record, but with a much higher density. A "pit" created on the raised portion of the track represents a digital 1, and a flat area (or "land") represents a digital 0. A master disk is used to duplicate the information by "stamping" the information onto other disks.

The stored data is read using a low-power laser: pits and lands on the surface of the disk reflect light differently. This difference in reflective quality is detected, and translated into readable data. CD-ROMs are largely used to distribute and reference large amounts of relatively static data such as on-line encyclopedias, legal citations, and (of course) musical recordings.

WORM Technology

WORM (Write Once, Read Many) optical disk systems use 5.25" disks encased in a hard plastic cartridge (6" by 5.25" by 0.4"). Unlike CD-ROMs, information can be written to the disk by the end-user, but only once as the writing process causes permanent alteration of the disk surface. Like CD-ROM disks, WORM optical disk drives write information using a laser which burns pits into raised portions of a spiral track on the surface of the disk. Once a pit has been created, that area of the disk cannot be restored to its normal flat surface: thus information written to a WORM disk is permanent. Figure 1 shows a vertical cross section (along tracks and sectors) of a WORM disk.

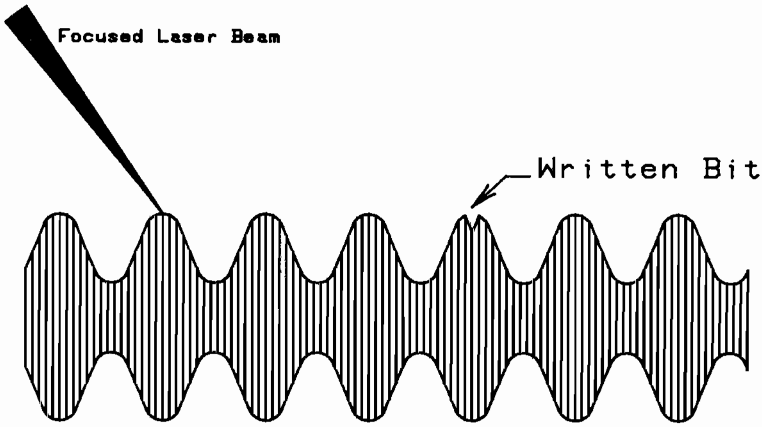


Figure 1: **WORM Optical Recording**

The same laser beam that is used for writing is used (at a much lower power) to read the information that has been recorded: each pit is interpreted as a digital 1, and each land ("no pit") is interpreted as a digital 0. Pits and lands are identified by the manner in which light is reflected off the surface of the disk.

WORM Drives and Directory Maintenance

WORM optical drives have one major "snag" that is not encountered with other mass storage technologies. Most existing file systems are structured so that some space on the medium is reserved for a directory. This directory must be updated each time a file is added, edited, or deleted. Since WORM optical disks cannot be rewritten, such directory maintenance is impossible. A number of solutions have been devised to overcome the WORM directory maintenance problem, some more successful than others.

One solution is to store data on the WORM disk cartridge, but use a separate flexible disk drive to store directory entries. Perhaps the most serious problem encountered with this solution is that the chances of hardware failure increase two-fold. Corruption of either the WORM disk or the flexible disk will cause a data loss. Another important factor is the added cost and inconvenience incurred by the requirement to maintain two disks for each data set. The directory disk and the WORM disk must always be kept together, as loss of one will render the other useless. Because of these problems, this solution is seldom used.

Another solution is to make the WORM drive emulate a standard flexible disk drive system. This is done by copying changed files to new sectors of the WORM disk, along with a copy of the entire directory structure. The drive then accesses the new directory structure to locate files. The advantage of this solution is that it is simple to implement, and fairly portable. Only a device driver need be written for each different type of computer system; files are accessed using the standard operating system commands. The obvious disadvantage to this solution is wasted space. Up to 50% of the disk space is used just to maintain the directory structure for an active file system. Furthermore, there is no standard access to older versions of a file.

A third solution to the directory maintenance problem is to use a series of linked directories to access files. When a file or series of files is closed, a directory is created for just those files. This partial directory can be maintained in several different ways. Some implementors store each partial directory along with the associated data. As a file or series of files is closed, a "link" (stored in the last directory entry) points to the location where the new data and new partial directory will be stored. Other implementors store data at one end of the disk and directory entries at the other. As files are updated, the data and directory entries grow towards each other. This type of solution reduces the directory storage overhead, and allows access to older versions of files. A disadvantage to this solution is that access time is increased, since the controller must access the disk several times as it follows the directory links while searching for a file. Also, developers must create special software for each type of computer system that uses their WORM disks since standard operating system access methods cannot be used. Since WORM drives are used mainly for backup and archival applications where access times are not as critical, the linked directory solution is generally more than adequate.

Although there are disadvantages imposed by the write-once limitation of WORM drives, they do have one unique advantage. Once data is written, it cannot be altered. This characteristic makes WORM drives excellent for storing information that must be maintained for legal and audit considerations.

Erasable-Optical Technology

Erasable optical disks have only recently been introduced as a viable mass storage technology. Several manufacturers, including Sony, Ricoh, and Hitachi, are now shipping erasable optical systems. A host of smaller companies will be introducing units within the year. There are currently three technologies associated with erasable optical disks: magneto-optical, dye-polymer, and phase-change. Magneto-optical is the only technology that has reached the production stage, as various problems with the other technologies will require further research before they can be made into marketable products.

Magneto-Optical

Magneto-optical technology, as the name implies, uses a combination of lasers and magnetic field effects to store and retrieve data. The disk is composed of a magnetic material, highly stable at room temperature, encased in a plastic cartridge. The value of a bit depends upon whether its magnetic orientation is "north-pole-up" (representing a value of 1) or "north-pole-down" (representing a 0). This is illustrated in Figure 2.

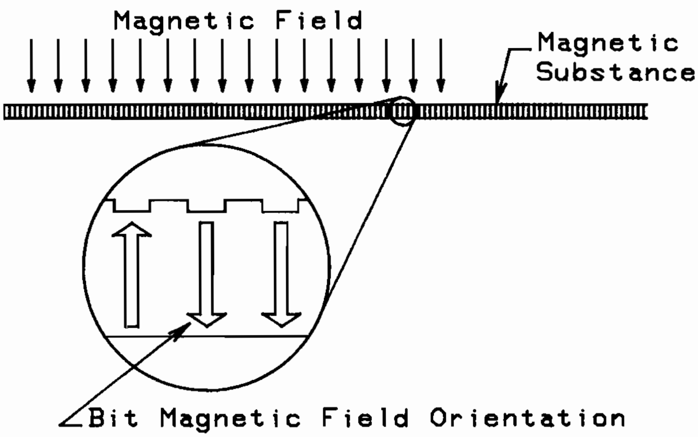


Figure 2: **Erasable Optical Recording**

A blank Magneto-Optical Disk (MOD) cartridge has all of its bits pointing north-pole-down. A magnetic coil in the drive produces a magnetic field that points north-pole-up. The strength of a magnetic field required to change the orientation of a bit varies with temperature: at room temperature, the magnetic coil is too weak to induce such a change (it is estimated that, at room temperature, a two ton magnet would be required to change the data on an MOD cartridge). However, at temperatures above 150 degrees Celsius (300 degrees Fahrenheit), the force required to change the magnetic orientation of a bit falls to almost zero, so bits are easily "flipped" by the magnetic coil. To write to the disk, a laser heats a spot on the disk to above 150 degrees Celsius, at which point the magnetic flux can easily be changed by a magnetic head. After the disk cools – only microseconds later – the magnetic flux once again becomes nearly impervious to magnetic fields.

The properties of the Kerr effect are used to read data stored on an MOD cartridge. The Kerr effect states that light will rotate in a particular direction if influenced by a magnetic field. An MOD drive uses this effect by directing a low-power laser at the surface of the disk. The light reflected from the surface will rotate in a clockwise or couterclockwise direction, depending upon the orientation of the magnetic flux of the surface. The read head detects the rotation direction, and sends a corresponding value of "0" or"1" to the computer.

One disadvantage of MOD cartridges stems from the fact that MOD systems must write zeros to the surface before data may be written to that spot. This means that the disk must rotate twice to complete a write operation: once to write zeros, and once to write the desired information. This quirk effectively increases the write-access time of an MOD drive by 40% over read access times.

Phase-Change and Dye-Polymer

The phase-change and dye-polymer technologies have similar advantages and disadvantages, though the technologies are slightly different.

A Dye-Polymer optical disk has two layers, each of which is treated with a different organic dye. Each dye absorbs light in a different range of frequencies, so either the top or the bottom layer can be selectively heated by using lasers of different frequencies. To write to the disk, the bottom layer of the disk is heated: it thermally expands upward, deforming the top layer and forming a stable bump. To erase information, the top layer is heated using a laser of a different light frequency which, in effect, "melts" that bump flat again. Information stored on a dye-polymer disk is read using a lower power laser to reflect light off the surface of the disk. The light reflected off a bump (interpreted as a binary "1") is different from that reflected off a flat surface (a binary "0").

A phase-change optical disk has a metal layer sandwiched between two layers of plastic. A high-powered laser is used to heat a spot on the metal layer, changing its molecular structure to a crystalline state (representing a "1"), and back again (representing a "0"). During the read cycle, the laser is directed at the spot using a lower power setting. An amorphous (non-crystalline) spot reflects light with a different intensity than light reflected from a crystalline spot. The bit-state is determined by a photocell, which detects differences in the reflected light.

The dye-polymer and phase-change technologies have encountered a few problems. First, a specific location on the disk can be written only a limited number of times — typically less than 10,000 — which is inadequate for most rewritable applications. Second, the drives use two different lasers (one for writing and one for erasing), making a drive using dye-polymer or phase-change disks more expensive than one using MOD cartridges. However, dye-polymer and phase-change disks do share one prospective advantage over MOD cartridges. Since WORM disks are read using differences in reflected light (as are phase-change and dye-polymer disks), a drive that can read both erasable optical and WORM disks is a distinct possibility. Furthermore, the media are generally less expensive and less susceptible to the environment than MOD cartridges.

Applying the Technologies

Both WORM and Erasable optical disks share a number of advantages over other storage techniques. These include the relatively long archival life of optical disks, the compactness of the optical disk cartridges, high capacity, and ease of data access. Optical drives are also less vulnerable to problems caused by surface contamination. Optical disks can hold at least 400 Mbits of information per square inch, compared to 49 Mbits per square inch for magnetic media. The removable optical disk cartridges are compact and (unlike high capacity Winchester hard disks), easily transported from one machine to another.

One of the greatest advantages that optical disks have over magnetic drives is that optical disks are much less prone to head crashes, which can result in the loss of great amounts of data. Magnetic heads must be very close to the disk to work properly: they are typically only 8 to 10 micro-inches above the disk surface. When shock or vibration occurs, the possibility of a head crash is very real. Optical heads, however, may be as far as 2 millimeters away from the optical disk surface, making a head crash extremely unlikely.

Though optical disk drives have a number of advantages over other storage methods, they are not completely free from drawbacks. Speed is an important consideration in any application. Though optical disk drives are faster than sequentially accessed magnetic tape, their access times are at best comparable to magnetic hard disk. When speed is crucial, the optical disk drive may be outperformed by magnetic hard disk.

Even though they share some features, the unique aspects of WORM and Erasable optical technologies make them particularly suited to very different tasks.

WORM Optical Applications

Due to their large capacities and write-once nature, WORM drives are best suited to applications that require large storage requirements and relatively rare updates, such as archiving and making backups. Because WORM optical disks cannot be rewritten, they are also the medium of choice for applications that require an audit trail.

Back-up

The huge capacities of WORM drives make them especially useful for backing up on-line mass storage devices. 9-track reel to reel tape and cartridge tape drives have capacities of less than 200 MBytes, while WORM drives have capacities ranging from 300–400 MBytes per side. Such large capacities make unattended backups feasible, reducing labor costs. When restoring a single file or a small group of files, a common operation in backup applications, access time is important. A WORM drive is a random-access device, and can access any part of the disk in under 200 ms. Tape systems, however, have to wind the tape to the desired section — an action that, depending on the length of the tape and the speed of a drive, can take anywhere from 10 to 100 seconds.

Archival

Data is archived because it is no longer needed on-line, but may need to be accessed at a later date. Archived data is never changed: only new versions are added. This characteristic provides a perfect application for the write-once nature of WORM optical drives. Examples of data that are typically archived include old source code and document revisions, seldom used software, and engineering drawings. The most important characteristics of an archival system are its media life and its storage capacity. WORM disks are particularly well-suited to archival applications because they exceed all other mediums in these characteristics.

Magnetic tape, for instance, has traditionally been chosen for the archival task. But magnetic tape has a relatively short life span: three years or so. Stretching or breaking of the tape and print-through (which occurs when the magnetic field from one layer of the tape migrates to an adjoining layer) contribute to the degradation of the medium. Optical disks, on the other hand, have a projected archival life of 10-20 years.

Microfilm, which has a 100 year life expectancy, is another medium commonly used for archival purposes. Microfilm's greatest disadvantage is its inconvenience: retrieving stored data is a tedious task. This characteristic makes microfilm great for storing information that is rarely (if ever) accessed, but too inaccessible for storing data that needs to be used. Optical disks offer an alternative that allows fast and easy access to any stored piece of information.

Audit Trails

There is frequently a need to keep a permanent record of a large series of transactions. Access to these transactions might be needed later for audit purposes. Some typical examples include maintaining credit card transactions, sales orders, maintenance records, and legal filings. Like archival storage systems, audit-trail storage systems need a large storage capacity and a long media life. In addition, some organizations require permanent, unalterable records of these types of transactions for legal purposes. The WORM optical disk, like no other media, fills all of these requirements perfectly.

Erasable-Optical Applications

Erasable optical disk systems have roughly the same capabilities as WORM disks; capacities, access times, and media life are comparable. Therefore, they can be used just as effectively in most of the same applications as WORM drives, except for audit-trail (which requires the unalterable nature of WORM media). The erasable nature of these drives, however, makes them much more flexible. Disks containing obsolete backups and archives can be reused, standard system software can be used with read/write access, and file changes can be stored much more efficiently. Since the disks are rewritable, they may be used in other applications for which a WORM disk would be unsuitable. For such applications, the erasable-optical system is a replacement or supplement to on-line storage such as a Winchester hard disk or flexible disk drive.

Archival

For certain archival applications, an erasable-optical disk performs far better than a WORM disk. Since erasable-optical disk drives "look" like a Winchester hard disk, files can be stored and accessed without using a special archival format. Access to these files is extremely easy; the disk need only be inserted into the drive and mounted. Files on WORM disks, on the other hand, must be converted from their archived form (by copying them to on-line storage) before they can be accessed.

This type of archival might be used, for example, to test software systems on several different versions of an operating system. Erasable optical disks could be used to store (and boot, as necessary) different revisions of an operating system. After testing was completed on one revision of the operating system, the next revision could be very easily installed just by changing optical disks and rebooting. In a similar manner, large, separate databases can be swapped in and out of on-line storage as needed.

Direct Access Storage Devices

The combination of large storage capacity, removability, and relatively fast access times makes the erasable optical disk a viable alternative to buying many, much more expensive, winchester hard disks.

In some systems, the erasable optical disk can completely replace the Winchester disk. It is estimated that 75 to 80% of workstation applications are low input applications which access the disk only a few times per hour. With such applications, the erasable optical disk is ideal and is already being used. The "Next" computer, for example, comes equipped with an MOD drive as its only mass storage device.

However, access time is critical in some common applications, such as in virtual memory management systems. For these types of applications, the erasable optical drive is not suitable: it can be used as a supplement to at least one Direct Access Storage Device (DASD), but not as a complete replacement.

Market Trends

As familiarity with optical disk technology increases, new and more efficient ways to utilize the features offered with this mass storage technique will be developed. Current innovations, such as jukeboxes and integrated systems, will increase in popularity as end-users realize that optical technology is a viable field, and not just a passing fancy.

Optical Jukeboxes

The optical jukebox, or autochanger, uses a mechanical arm to swap either erasable-optical or WORM disks from a large library to usually one or two on-line optical drives. Jukeboxes are used when massive storage capacities and an access time of one to two seconds are required. These systems typically have total storage capacities in the tens of gigabytes, with on-line storage accounting for 5 to 15% of the total capacity.

Integrated Systems

A major innovation to look for in the future will be the widespread use of optical disk drives in integrated mass storage systems. These systems use a combination of Winchester hard disks, optical disk drives, and tape drives to store files. The particular method used to store a file is dependent upon the frequency of its use. The file will automatically migrate from fast on-line Winchester hard disk storage, to optical storage, and finally to tape storage as the frequency of its use decreases.

The Optical Market Share

The WORM drive is currently well established in the market place. Rothchilds Consultants estimates that 100,000 WORM drives were sold in 1988. By the end of 1990, they estimate that this figure should reach 300,000. According to their figures, only 40,000 Erasable-optical disk were sold in 1988, the first year it was introduced. By the end of 1990, this number should rise to 900,000. These estimates for the growth and applications of WORM and erasable optical disk drives are shown in Table 1.

	Estimated Drives in Use					
	WORM			Erasable		
Application	**1986**	**1988**	**1990**	**1986**	**1988**	**1990**
Hard Disk Complement	1,600	60,000	50,000	0	5,000	70,000
Hard Disk Replacement	0	0	0	0	12,000	260,000
Archival/Backup (off-line)	100	25,000	50,000	0	2,000	40,000
Save/Restore (on site)	0	0	0	0	16,000	460,000
Digital Image Capture	1,000	15,000	200,000	0	5,000	70,000
TOTAL:	2,700	100,000	300,000	0	40,000	900,000

Table 1: **Estimated WORM and Erasable Optical Disk Usage**

These numbers reflect the growing popularity of optical storage technology. The numbers for WORM drives sold continue to rise due the maturation of the technology. The erasable optical disk drive, however, is much more flexible than the WORM drive, and costs for both systems are roughly equal. These factors should eventually lead to the replacement of the WORM drive by the erasable optical drive in all but niche markets where non-erasability is important.

One of the reasons for the growing popularity of erasable optical systems is that they fill a large gap between slow, inexpensive tape systems and fast, expensive Winchester disks. Current computer systems need a large and ever growing on-line storage capacity. Although capacities continue to increase and prices decrease for Winchester systems, they cannot compare to removable erasable optical disks. System managers will continue to equip their systems with at least one Winchester disk drive into the future, but will increasingly turn to the erasable optical disk for supplementary on-line storage.

A CASE STUDY FOR NETWORKING PRODUCTION, ADMINISTRATION AND SALES

HEWLETT PACKARD

Rainer Schulze
Herrenbergerstr. 130
D-7030 Boeblingen
West Germany

The Example

ANGLIAN WINDOWS in Norwich, England is the market leader in double glazing and home improvements - particulary conservatory-type home extensions. In 1987 ANGLIAN overtook its nearest rival and became the leading power in the market of double glazing as a result of the right network concept. This means about $ 3M turnover per week for ANGLIAN.

Data Processing three years ago

What is behind this concept and how could it be applied for other companies?
Some years ago, an ICL central computer was responsible for the order processing in Norwich. All orders from the 48 branch offices and 140 showrooms were sent on paper to Norwich, entered into the computer where the data were processed in batch, printed on paper, evaluated and then sent back to the offices. The consequence of any following change order from the customer was that a piece of paper had to be sent to Norwich and back to the customer again. This caused a delay. Furthermore, the cost invoices and factory scheduling could only be done once a week. Five years ago, the first migration to MPE/V mini computers (S70, S68, S58, S48) took place. This allowed a distribution of program packages onto different computers which caused a considerable increase in data processing throughput. One year ago, local networks were built up in the branch offices, tested and 6 months later put at the users' disposal.

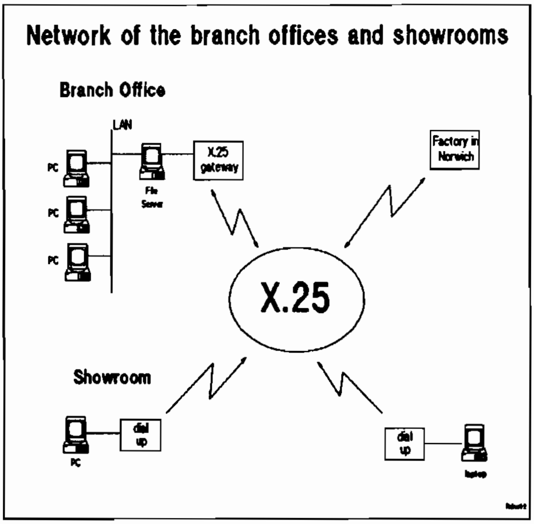
Status of the current automation

General

This year the last step was concluded by migrating and switching all four MPE/V systems onto two MPE/XL computers (S950). Automation is now applied to each sector of the company: sales, administration, production, development and delivery. The information exchange within the company now takes place via HPDESK. Nearly all employees can receive and pass on news very quickly and at the same time. Future growth is possible by upgrades to a S955 or S960. X.25 has been chosen as the network transport method. Its popularity in Europe allows easy growth into the other European countries.

Sales

Each of ANGLIAN's branch offices has been equipped with a LAN to which MS-DOS PCs and a file server are connected. PCs can now access the HP3000/950 mini computers (LAN-HOST) by means of a LAN file server and an X.25 gateway. The PCs can also communicate with each other via the LAN (Figure 1).

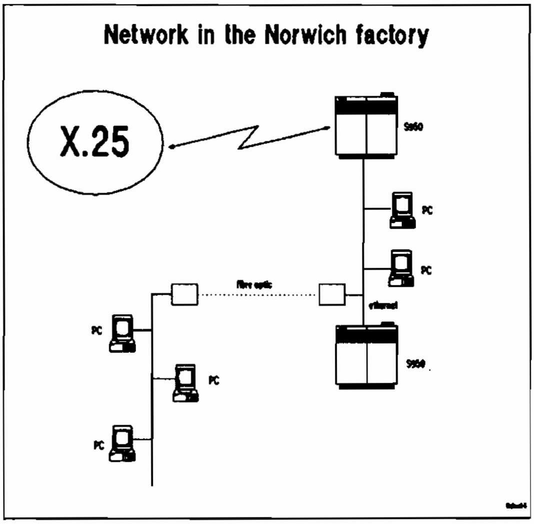


Take, for example, a customer who wants to buy windows from ANGLIAN. An ANGLIAN salesman visits the customer at home, presents the range of products on the screen of a laptop (2400 different designs are at his disposal) and enters the customer's order as well as the window measurements in the Portable. Sometime during the day, the salesman transfers the data from a showroom or branch office via X.25 to the HP3000. It is also possible to transfer the data directly from the customer's site by means of a modem.

If necessary, the company's designers consider the individual customer's desires as to window forms and design, and accept an order for development and verification. These employees then develop the desired forms in a very short time by means of workstations (CAD/CAM), and then transfer them to the factory in Norwich for further processing .

Administration

The data transmitted via X.25 is processed in the factory by two HP3000/950s. Here there are several program packages responsible for the immediate processing of the order and construction data. For example, one job optimizes glass cutting and decreases scrap by cutting as optimally as possible. This is also done for all other materials such as wood, plastic (PVC), etc. Under MPE/V on a Series 70 the job took about 10 hours. It only needs 50 minutes under MPE/XL. Four years ago, the cutting of raw material could only be calculated once a week; one year ago, this was done once a day. Now the computer is able to calculate this cutting three times a day. Furthermore, automation allows an exact overview of which orders are processed when and where. (Figure 2)



Production

By means of barcode readers, the production data can be entered very quickly into the network and the corresponding cutting data for the glass and plastic are transmitted directly to the cutting machine.

Development

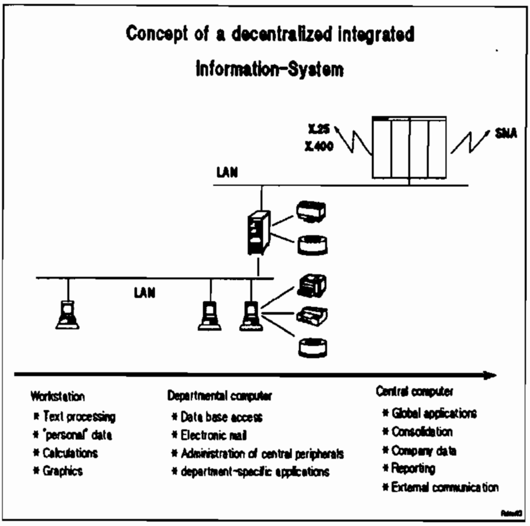
CAD/CAM workstations in selected offices use AUTOCAD in order to develop individual orders. These individual construction plans are then transmitted to the development departments in the factory where within 24 hours - compared to 3 to 4 weeks some time ago - the order can be verified and confirmed. Also any modifications to the order are possible in a very short time frame.

Concept of a decentralized network

Many companies' organization has changed in the last few years through the decentralization and expansion of these companies due to the market changes. The challenge today os to react quickly to market changes , in the organization and in the technology. Company growth and a more acceptance of customers' wishes represent two vital aspects.

In the future the market will be marked by short development times of new products and the development of products and services which require a fast reaction from every competitive producer.

ANGLIAN WINDOWS is a typical example for concerning themselves with the challenge and the future oriented market character. ANGLIAN created a decentralized concept where company networking was standardized thereby making it possible for the offices to communicate with each other and with the factory. Here the basis is represented by the PCs together with efficient departmental computers. Therefore the computer performance is offered at the place where the user really needs it. The first step to bring computer performance closer to the user the user is to use PCs as workstations followed by departmental computers being the link between workstations and central computer (Figure 3). The tasks between departmental and central computers can be taken over partly from each other depending on the demand and need.



Workstation

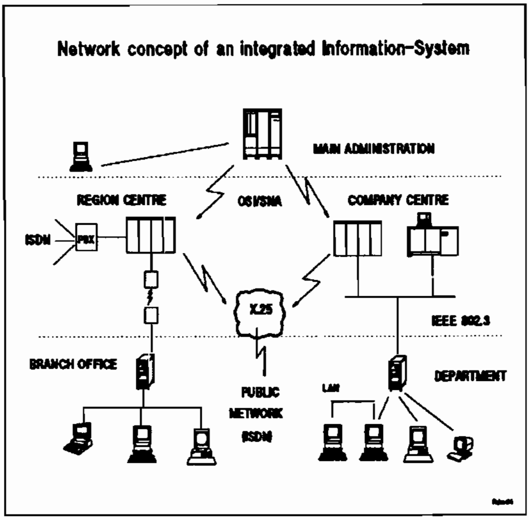
If the user has to access a specific amount of data at his desk (e.g. word processing, graphics, calculations, etc.), a PC is used instead of a standard terminal because the PC is more user friendly and provides a predictable response time through local processing.

Departmental computer

The departmental computers are used as file and/or database server, i.e. their task is to provide locally the data for the working group or for the department. Furthermore, the departmental computers are being used by the application programs which carry out the department-specific tasks and manage departmental peripherals.

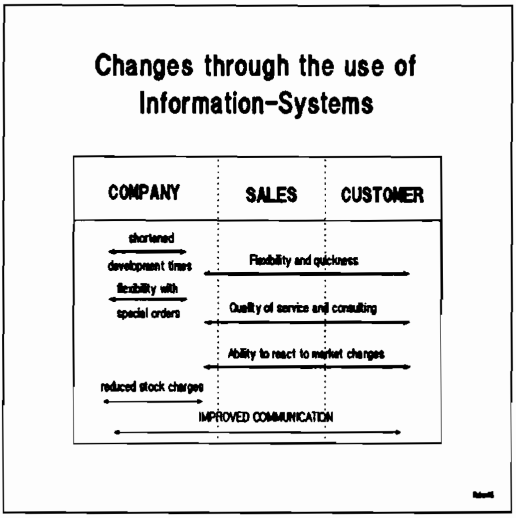
Central computer

As a result of implementing PC workstations and departmental minis, the central computers need only to contain company-wide databases such as financial accounting and salary administrations applications (Figure 4).



Conclusion

The concept of ANGLIAN is to combine many mini computers with decentralized distributed PC networks in order to distribute the tasks of sales, administration and production. Then we do not rely any more as before on the total task completion on only one central computer. This concept guarantees that the branch offices form autonomous sales and profit units which in spite of high self responsibility can work closely together with the main administration if required. This results in very fast turnaround through the detailed knowledge of the need of raw materials, an essentially increased quality of service and consulting, the ability of reacting quickly, e.g. as to market changes, a high flexibility as to special orders and all in all an improved communication, e.g. between each single department. (Figure 5)



The possibility to react quickly to individual customers' requirements and ideas will be considerably important in the future. The example of ANGLIAN shows that the equipping of the salesman with laptops and the corresponding linking of these portables into the network makes this flexibility possible.

This example easily applies to other company branches. Just imagine to use workstations or laptops in the furniture industry, especially in the kitchen furniture sector. For example, a customer who wants to buy a standard kitchen could be consulted at home (laptop) and could immediately conclude the contract there. In a showroom, his kitchen equipment could be arranged with the help of a workstation according to his individual requirements and the given space in his kitchen. Special designs would be usual and not only an exception.

This concept also applies to the clothing industry. Individual designing of dresses or suits would be no major problem any more. There are many more industries like this and the list could easily be continued.

For the construction and successful introduction of such an Integrated Information System Concept, it is important that users and suppliers of these systems work together very closely and it is necessary that the supplier gives competent advice to the user. The operating sequences of a company are subject to permanent change, e.g. through expansion or through further technical changes in different areas. A well-planned strategy largely covers such changes but on the other hand it must be adapted to changes in the company structure if needed. That's why the concept must be flexible, must be supported by standards as well as the concept must be able to easily integrate any change. This is the case with this example.

WAYNE McKINNEY

LOCKHEED ENGINEERING

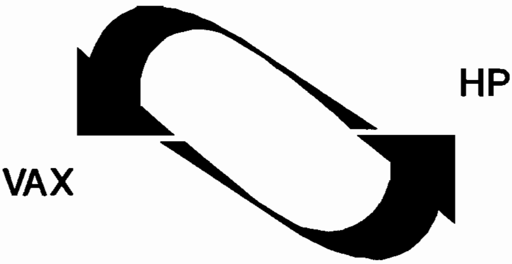
&

SCIENCES COMPANY

2400 NASA ROAD 1

HOUSTON, TEXAS 77058

713 333-6258



A USER'S VIEW OF HP/VAX CONNECTIVITY

7011-1

I have often heard it said that in getting a job, it's not what you know but who you know. Well, the same thing can be said about computers today. Though one computer might store a great deal of data and serve its customer's information needs, sitting isolated it serves only those customers who can log into it. Those who use foreign computers have no way of accessing the information stored in this other computer.

I found myself in a situation where my department's minicomputer was isolated from the other computers that my company owned. Though some employees were restricted to logging only onto my department's computer, they still wanted to send and receive electronic mail messages to and from my company's other computers. For several months, I worked at solving this connectivity problem. What follows is a description of both the research I had to do and the problems that I ran into.

By now you may be wondering who I am and what it is that I do. Well, my official title is "Computer Programmer Associate". I know that still doesn't tell one too much, it does bring up the point that I do not have the title "Network Manager" or "Data Communications Specialist". Though I am a programmer, I am also a user of computer services.

As a "user" I had very little previous experience with data communications. I had sent files back and forth between PC's and PC's and VAXes, but this project went beyond my previous experience. In a way, this lack of experience helped because I had nothing to sway my research in one direction or another. We chose as one of our first goals to define what we meant by "connectivity".

For purposes of this project and for the presentation, "connectivity" was defined as the "ability to share data among computers from the same vendor or from different vendors." With such a general definition, we found that we could keep our minds open to the varied solutions that vendors proposed.

Implicit in our definition of "connectivity" is the method of "sharing data". We are not talking about transferring data to a floppy disk or to a tape and then transporting that tape to the receiving computer and then loading the data onto it. While this method accomplishes our objective of sharing data, it is not "true" computer connectivity.

"True" connectivity, in our situation, was the ability to transfer data on demand to another computer, either through connecting directly to one of its ports or indirectly by connecting to its network. With this in place, both supervised and unsupervised transfers of data could be accomplished.

With this groundwork down, it would now help to describe in a little more detail what the situation was like with our HP and VAX. My department operates an HP 3000, model 70, while another department within Lockheed operates a DEC VAX 8650. As the situation now stands, our HP and VAX are on separate networks. The VAX is on a large company wide network that operates under the DECnet network protocol. DECnet is very similar to Ethernet, but with enough differences to make it impossible for it to operate in conjunction with our HP's computer network which uses IEEE 802.3.

As one can imagine, if two computer systems are basically isolated, then data would be rarely shared. Even as these two systems now stand, the only data sharing going on between them is one way. Our HP builds a data file that can be uploaded to the VAX only when an automated session from the VAX logs onto our computer as a user. A normal session upload can occur using Kermit. This, however, is an isolated example. Data is simply not shared to any great extent between the VAX and HP.

On a previous VAX network that I was on, it was possible to start a new session or logon on one VAX while maintaining a session on the other VAX. This feature was quite handy when one wanted to stay in a session on one VAX, pop over to the other VAX to find out something, and then return to your original session. Such a situation was simply not possible between the VAX and HP because they were on separate networks and their networks were incompatible.

The most important weakness in the present situation was that it was impossible to share electronic mail messages between both computers. The VAX has its All-In-1 format and HP has its HP Desk format. Both are true electronic mail messaging systems, but their formats are incompatible. Even if the two computers were connected together, there would be no way for one computer to correctly receive and display mail sent to it from the other computer.

If it looks like all of this is leading up to requirements you are right, for the last weakness in the present situation became the starting point for one of our primary requirements. That requirement was that there must be a mail interface implemented in any system that connects these two computers. The mail must not only be able to pass between the computers, but it must also be translated to accommodate the format of the electronic mail system in the other computer.

Expanding on this requirement is the additional requirement that there be implemented a single mail box. This means that a user doesn't have to remember to log onto his VAX account if he is presently on the HP. Any mail sent to him on the VAX should be directly accessible to him on the HP and the reverse should be true also.

Because our HP has a wealth of information, another requirement was that data uploads and downloads should be possible between the two computers. This would be in the form of files that could be sent back and forth.

In addition to this requirement is the requirement that it must be possible to have virtual terminal access from either computer. This means that if you want to jump onto the VAX while you are in your HP account, you could give a command to start a new session on the VAX. When you are through there, you would still be in your HP session. The reverse must also be possible.

Looking to the future, we saw as a final requirement that this system be modifiable. No solution is ever truly permanent because situations themselves are always in a state of change. What we didn't want was a network that remained static while everything else around it got upgraded and changed. That network must be programmable and easily reconfigured to meet the requirements of changing conditions. Our investment would be maximized because this network's useful life span would be much longer than if it were not modifiable.

Without any formal training in data communication, I found that the research that I had to do even to find possible solutions to this connectivity problem was, at first, staggering. There seemed to be so many ways to accomplish such a connection. Gradually, however, it became clear that there was both a hardware and software side to any solution.

The hardware side dealt with how the physical connection between these two computer would be accomplished. A little research showed that most vendors were aimed at two possible solutions.

One solution was to implement a bridge between two computers. Though the networks would still be isolated, any message from one computer that was directed to the other computer, would be routed away from its originating network and over to the destination computer's network. The second solution involves many of the same ideas but it is implemented through a direct connection. A bridge would not be needed because both networks would be compatible. Messages could pass back and forth unimpeded by hardware differences.

The software side of the vendor's proposals looked very much like both of the hardware solutions. One solution was to provide for a means of translating the messages when they pass between the networks. In this case, the All-In-1 mail would translate to HP Desk format and vice versa. Any differences would be resolved by the software.

The second software solution was implemented using software that placed these messages in a standard format. One format that was growing in popularity at the time of my research was that of the X.400 standard. Even if the messages originated on completely incompatible computers, the receiving computer could still understand the message sent by the originating computer because the format of the message followed a standard format that both computer manufacturers agreed to. No translation would be needed when the messages are sent.

Speaking of vendors it would only be fair to mention the names of the ones I talked to. These included the Hewlett-Packard, the Wollongong Group, the Apex Group, Digital Equipment Corporation, Minisoft, Forest Computer, and Soft-Switch. These vendors were not listed in any special order. Even if they were it wouldn't matter because each one of them (except for DEC) offered a solution very much along one of the lines of those I described.

I found that each vendor had good products and each employed knowledgeable salespersons and technical representatives who were quite willing to put up with my "stupid" questions until I finally understood what they were trying to get across to me. On the other hand, don't let a vendor intimidate you. What you should be looking for is the expertise that you need to achieve your connectivity goals. A vendor's services are usually cheaper than those of any full time data communication specialist whom you might hire to implement a similar connection.

When it was time to make a recommendation, I chose the vendor who proposed to implement a direct connection between the VAX and HP. They also proposed to implement message translation between these two systems.

As it turned out, the V/Delta 5 version of HP's MPE operating supported Ethernet communications. With a direct connection now possible, we would now have a simple implementation because very little complicated hardware and extra software modules would be involved to make this connection functional. Reliability would be greatly increased at no great expenditure of resources.

You may have noticed by now, but I quite often speak of this connectivity in the past tense. That's because even to this day, my connectivity project is no longer active. The reasons for this are quite simple and they are illustrative of what one might run into when trying to create a connection between two computers owned by different groups.

The first problem was that of cost versus use. While a great number of people use the VAX's All-In-1 mail system, very few people use our HP's HP Desk system. This is not due to any lack of functionality in HP's implementation of HP Desk; rather its based on the fact that far more people have access to All-In-1 than to HP Desk. HP Desk is only on two computers - All-In-1 can function from almost any VAX that Lockheed and our NASA customer own.

What's more, there turned out to be only a few managers who could not log onto the VAX but who could log onto the HP. Even with increased versatility that this connectivity would bring, it was simply not worth the expenditure to satisfy the needs of only a few managers.

Another reason for this project not completing was that of security. With our HP isolated, security on it was very tight. No one could get on to the HP who didn't have an account and even if they did steal someone's password, our HP automatically takes them to a menu from which they cannot escape to a colon prompt - thereby getting access to files. The VAX was another story.

Because our VAX was connected to a large network, it was constantly in peril of getting broken into. If there was any way that some outsider could break into the VAX and then use our data communication setup to steal data from our computer, then all of our information would be in danger.

As both a programmer and a user, I learned a great deal from this project both about data communications and in dealing with vendors. It is possible for the average programmer/user to research the solutions available to achieve connectivity and maybe even to bring that solution to reality. That programmer, though, must be both careful and wise for it's not who you know in this business, but what those people know that can make or break any project involving computer connectivity.

ATP PORT SOLVES TELEX OPERATOR PROBLEM

D.R. Kambey & Richard Siregar

P.T. ARUN NATURAL GAS LIQUEFACTION CO.
Tromol Pos 22, Lhokseumawe
Aceh Utara, Sumatera
Indonesia

ABSTRACT

In an effort to achieve higher productivity and reduce the human intervention on data entry, with minimum cost, many data processing organizations are trying to automate the transfer of data from one location to another.

Today, there are a lot of products available in the market that are used to transmit and receive data between one computer and another. But, the organization that wants to utilize this technology, must evaluate in detail each product offered by the vendor. These products could be software, hardware and data communication devices.

Many companies have their own telex machine as a device to communicate with another company e.g. send product descriptions, purchase orders, requisitions and other correspondence. This information has to be accurate, especially for the technical specifications. Without double checking we can not ensure,whether the information is correctly entered by the operator.

This paper details P.T. Arun experience in sending material descriptions interactively from the HP3000 data base to the Mobil Oil telex network, by utilizing the HP3000 ATP ports connected to the telex machine and a custom made program as a protocol.

This system has been running for nearly four years with failure rates less than 10 percent.

1. INTRODUCTION

PT. Arun NGL Co. is a natural gas liquefaction plant situated in the province of Aceh on the Northern tip of the island of Sumatera, Indonesia.

LNG is an acronym of Liquefied Natural Gas. The liquifying principles is to reduce the gas temperature from 174 C above zero to 160 C below zero. This is meant to enable transportation and storage because the volume of the gas before and after it was liquefied are compared as 600 : 1.

Different from other energy resources, LNG product is a long term business that integrates producer, consumer and the carrier service.

Long term sales contracts of the LNG product of Indonesia have been signed by PERTAMINA (Indonesia State Oil Company) with Japan and South Korea as buyers.

PT. Arun was given a responsibility by PERTAMINA to meet this agreement. For this purpose PT. Arun maintains all assets including the high technology equipment that supports the plant operations. Without a good system, it is difficult to control the availability of spare materials and replenishments for the related inventory.

Some of the materials are not available locally and have to be purchased abroad. Placing an order to the manufacturer is an important part, as the specification has to be accurate. Wrongly entered material specifications result in wrong delivery, and creates delays on the equipment for repair.

Good material specifications kept on the Data Base become meaningless, if some parts of the transactions are still processed manually.

This paper stresses the process of sending PT. Arun's purchase requisition on-line from the HP3000 data base to a telex network.

2. MAINTENANCE MANAGEMENT SYSTEM OVERVIEW

MMS is a HP3000 based plant maintenance system integrating three important plant functions:

- Maintenance Planning & Scheduling
- Inventory Control, and
- Purchasing sub systems.

Maintenance Planning & Scheduling covers:

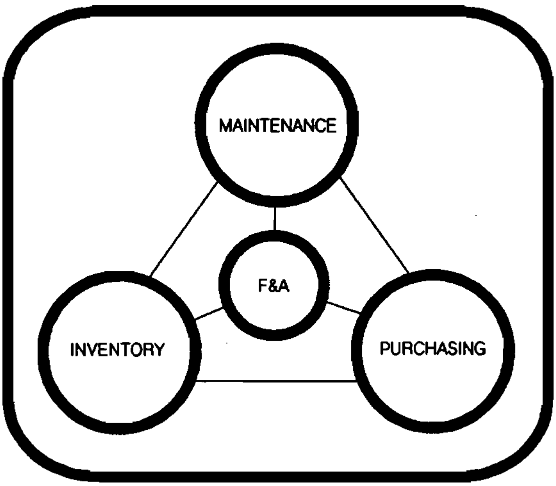
- Work request system
- Work order system (regular and preventive maintenance)
- Equipment Master and component cross reference
- Craft tables.

Inventory Control sub system covers:

- Stores Master.
- Stores transactions (Issue, received, allocation etc).
- Stores replenishment
- Stores equipment cross reference

Purchasing sub system covers:

- Vendor Master
- Purchase requisition
- Quotations
- Purchase order
- **PURCHASE ORDER TELEX**
- Expediting



3. PT. ARUN EXPERIENCE IN SENDING PURCHASE ORDERS

One responsibility of the Purchasing Department at PT. Arun is to fulfill the needs of the company to purchase material for warehouse stock, project materials and a non stock or direct charge materials.

Material requisitions are prepared by the originating department and approved by the appropriate management level (Figure 3-1). Requisitions are then sent to the Material Control Group for verification and entered on the computer (Figure 3-2). A computer print out is then produced and sent to the Purchasing department for further processing (Figure 3-3).

A purchase order (PO) is then sent to a selected vendor. For a local vendor, the PO's are sent by mail.

Historically, however, PO's for a foreign vendor were initiated by telex forms being prepared with the material requested and sent to the telex operator (Figure 3-4).

The telex operator then keyed in all the data line by line on the telex machine, and had it transmitted. A paper tape was also produced for correction or re-transmission.

This manual process was discontinued after the implementation of THE POTELEX Interface on September 20, 1986.

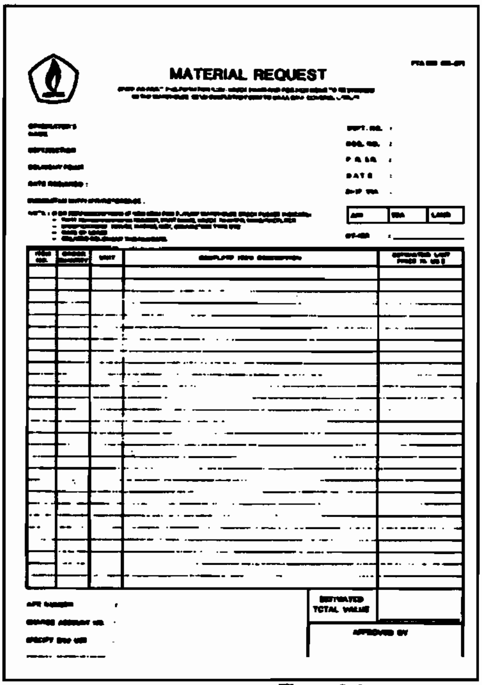

MATERIAL REQUEST

Figure 3-1

```
                       NON-STOCK P.O. REQUISITION ENTRY (51)                  MSK390
ACTION __                                                                     MPP596
REQUISITION NO 018708140___ DATE REQUIRED ______    TERMS ____________________
REQUISITIONER  __________   DELIVER TO    ______________
TAG NO         ____________ WORK ORDER NO ___________  USE ____________________
ACCOUNT  CODE  ______       A F E NO. ______      DEPT. REF. NO. ______________
AUTHORIZED BY  [__________] SHIP INSTR    ___________________    CURRENCY ___
SUGGESTED VENDOR NO. ________               TOTAL EST. VALUE ____________
             NAME ___________________________    ORIGINATOR ____________

A ITEM QTY/ST/PRI/TYPE         DESCRIPTION                      PRICE/CUR/MEAS
- ____ ______  ___________________________________________   _____________
        - - -  ___________________________________________   ___ ______
               ___________________________________________   ____.___.___
PIREC ______   ___________________________________________    MESC _____
- ____ ______  ___________________________________________   _____________
        - - -  ___________________________________________   ___ ______
               ___________________________________________   ____.___.___
PIREC ______   ___________________________________________    MESC _____
- ____ ______  ___________________________________________   _____________
        - - -  ___________________________________________   ___ ______
               ___________________________________________   ____.___.___
PIREC ______   ___________________________________________    MESC _____
```

Figure 3-2

```
REPORT MPR174                                                  PAGE      1
                   PURCHASE REQUISITION - STOCK               DATE  25/08/87

 REQ NO: 018708140   0001  P.O. NO: ___________     UNIT COST:     1,408.91

 ACCOUNT NO: 09-001   AFE NO:        DEPT NO: MN/MRG/0249       PRIORITY: D

 STOCK NO:    36-100-194       QUANTITY:          2       U/MEAS: EACH

 DATE REQUIRED: 25/11/87  DATE PROMISED:             STATUS: -OPEN

                                        EST. VALUE (USD):       3,500.00
DESCRIPTION:   SWITCH - DETECTOR, C/W EXPL. PROOF JUNCTION BOX.
               MFG SMITH METER SYSTEMS, P/N 36-650381-375


                        VENDOR INFORMATION

   CIN                                  169
                                        MOBIL OIL CORPORATION
                                        PURCHASING DEPARTMENT 3225
                                        GALLOWS ROAD
                                        FAIRFAX , VIRGININA 22039
                                        PHONE 710-833-0372

 SPEC INSTRUCT  _____________________________________

 REQUISITIONER  [__________]

 VENDOR#      DATE      QTY   UNIT   UNIT COST    P.O. #      REQ BY
```

Figure 3-3

PTA 10 (02 – 76)

PT. ARUN NATURAL GAS LIQUEFACTION CO.

OUTGOING TELEX / CABLE

IF PERSONAL CHARGE TO ACCOUNT OF :

ORIGINATOR : RSYAKRAN

APPROVED BY : Th for RS

TO : RNYFP – NEWYORK
SINMO – SINGAPORE

FROM : PTA – LXME

DATE : 14/09/86

CC : ZAHRI – MAINT

MESSAGE

//(BARLOW)(ROARK)(FERRANTE)/(ONG)(JEYA)/

ENTER PTAM61888/AIR/ROS 25 NOV 86

ITEM	QTY	DESC.	PTA S/N
1	2 EA	SWITCH-DETECTOR C/W EXPL. PROOF JUNCTION BOX MFG SMITH METER SYSTEM. P/N 36-650381-375	36-100-194

REFER TO: PREVIOUS ORDER PTAM 50346
YOUNG MAN & CUFF INC

REMARKS: EST. VALUE USD 3500.00.

Printed by pta

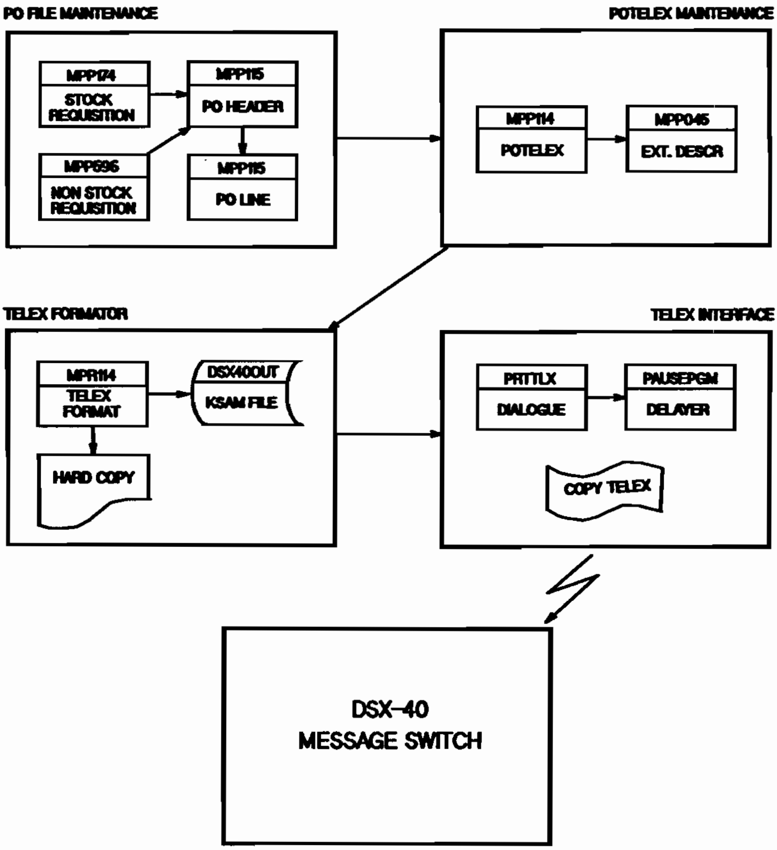
Figure 3-4

4. POTELEX INTERFACE

The POTELEX Interface is systematically divided into four parts:

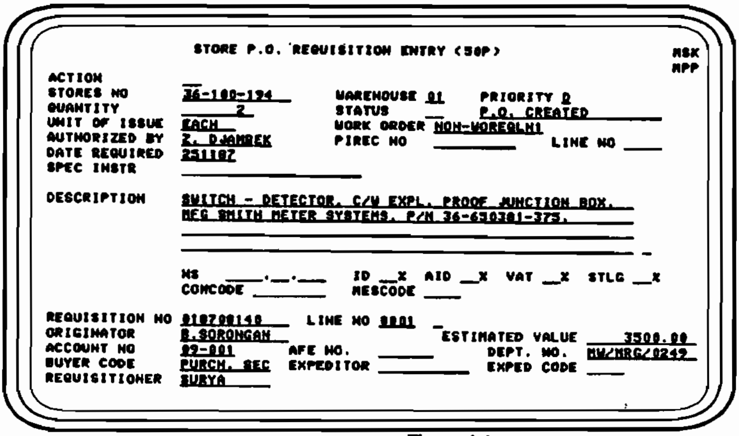
- PO File Maintenance
- PO TELEX Maintenance
- Telex formator and
- Telex Interface.

The diagram below shows the interrelation between each part.



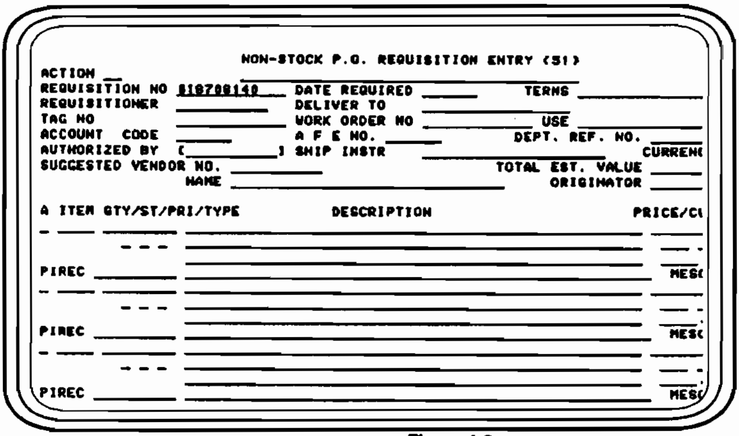
4.1. Requisition & Purchase Orders

Requisitions are created from this screen. For a non stock requisition, one requisition number may contain several line items. Stock items or warehouse materials contain one requisition number for each stock item (Figure 4-1 and 4-2).



```
                    STORE P.O. REQUISITION ENTRY (50P)                      MSK
                                                                            MPP
ACTION           __
STORES NO        36-100-194      WAREHOUSE 01     PRIORITY 0
QUANTITY                  2      STATUS __        P.O. CREATED
UNIT OF ISSUE    EACH            WORK ORDER NON-WOREQLN1
AUTHORIZED BY    Z. DJAMREK      PIREC NO ________         LINE NO ____
DATE REQUIRED    251187
SPEC INSTR       ________________

DESCRIPTION      SWITCH - DETECTOR, C/W EXPL. PROOF JUNCTION BOX,
                 MFG SMITH METER SYSTEMS, P/N 36-650381-375.
                 ______________________________________________
                 ______________________________________________ _

                 NS ____.__.___     ID __X  AID __X  VAT __X  STLG __X
                 COMCODE _______    MESCODE ___

REQUISITION NO   018700140       LINE NO 0001  _
ORIGINATOR       B.SORONGAN                      ESTIMATED VALUE    3500.00
ACCOUNT NO       09-001    AFE NO.    ______     DEPT. NO.    MW/MRG/0249
BUYER CODE       PURCH. SEC  EXPEDITOR ________  EXPED CODE   ____
REQUISITIONER    SURYA
```

Figure 4-1



```
                       NON-STOCK P.O. REQUISITION ENTRY (51)
ACTION __
REQUISITION NO 018700140   DATE REQUIRED ______      TERMS ___________
REQUISITIONER  _________   DELIVER TO    ______________
TAG NO         _________   WORK ORDER NO __________  USE ___________
ACCOUNT  CODE  ______      A F E NO. ______         DEPT. REF. NO. _____
AUTHORIZED BY  [_______]   SHIP INSTR    ______________          CURRENC
SUGGESTED VENDOR NO. ________                       TOTAL EST. VALUE _____
                 NAME ________________________            ORIGINATOR _____

A ITEM QTY/ST/PRI/TYPE           DESCRIPTION                        PRICE/CU
- ____ ________  ________________________________________________  ______
           - - - ________________________________________________  ____ -
                 ________________________________________________  ____
PIREC ________   ________________________________________________   MESC
- ____ ________  ________________________________________________  ______
           - - - ________________________________________________  ____ -
                 ________________________________________________  ____
PIREC ________   ________________________________________________   MESC
- ____ ________  ________________________________________________  ______
           - - - ________________________________________________  ____ -
                 ________________________________________________  ____
PIREC ________   ________________________________________________   MESC
```

Figure 4-2

After the vendor is selected, a buyer then creates a purchase order number, states when the material is required on site, and enters other detail information required.

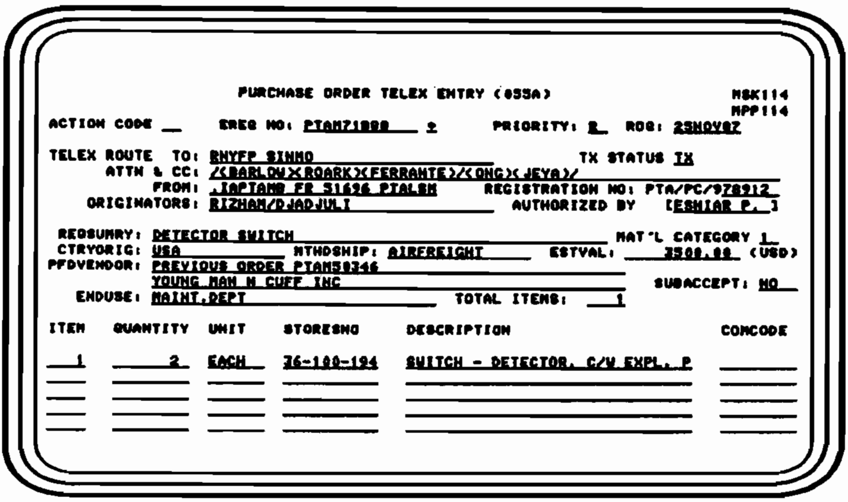
Figure 4-3 shows the line item information for a purchase order. Other similar materials to be placed on order with the same vendor can share the purchase order number on different line items.

The buyer does not need to retype the description, as some information is brought forward from the requisition.

```
                      PURCHASE ORDER LINE ENTRY (57)                     MSK294
 ACTION CODE __                                                          MPP294
P.O. NO        PTAM71888         P.O. LINE NO     1 _     PRIORITY D_
REQ NO         018708140         REQ LINE NO      1
STORES NO.     36-100-194        STATUS CODE     05
QTY ORDER             2          ACCOUNT NO      09-001
QTY RECVD             2          CURRENCY TYPE   USD
UNIT PRICE       1377.62  EACH   EXTENDED PRICE          2755.24
UNIT MEASURE   EACH              WORK ORDER NO   NON-WOREQLN1
PROM DATE      ______            TAG NO          ____________
REPROM DATE    ______            REQUISITIONER   SURYA
NO OF REPROM   _0                NOTICE          _
EXPEDITOR      __________        EXPED CODE      ____
SUB-VENDOR     ________          ______________
 DESCRIPTION   SWITCH - DETECTOR, C/W EXPL. PROOF JUNCTION BOX.
               MFG SMITH METER SYSTEMS. P/N 36-650381-375.
               __________________________________________________
               __________________________________________________ _

               MS    ____.__.___   ID 20%  AID __%  VAT 10%  STLG __%
               COMCODE 27650010    MESCODE 67__
     COMMENT   ____________________________________________________________
```

Figure 4-3

4.2. Purchase Order Telex



```
                    PURCHASE ORDER TELEX ENTRY (055A)                    MSK114
                                                                         MPP114
ACTION CODE __    EREQ NO: PTAM71000   ?     PRIORITY: R  ROS: 25NOV87

TELEX ROUTE  TO: RNYFP SINMO                         TX STATUS TX
     ATTN & CC: /(BARLOW)(ROARK)(FERRANTE)/(ONG)(JEYA)/
          FROM: .JAPTAMB FR 51696 PTALSM      REGISTRATION NO: PTA/PC/978912
   ORIGINATORS: RIZHAM/DJADJULI               AUTHORIZED BY   [ESHIAR P. ]

  REQSUMRY: DETECTOR SWITCH                              MAT'L CATEGORY 1
  CTRYORIG: USA          MTHDSHIP: AIRFREIGHT    ESTVAL:    3500.00 (USD)
 PFDVENDOR: PREVIOUS ORDER PTAM50346
            YOUNG MAN N CUFF INC                          SUBACCEPT: NO
    ENDUSE: MAINT.DEPT                     TOTAL ITEMS:   1

ITEM  QUANTITY  UNIT   STORESNO     DESCRIPTION                        CONCODE

   1        2   EACH   36-100-194   SWITCH - DETECTOR, C/V EXPL, P
```

After the foreign purchase order is created, a purchase order telex is then prepared from screen 55 above.

Action code "O" and entered PO number on EREQ NO field shows the purchase order line information. All mandatory fields should be entered before selecting action code "A" to create the POTELEX record on the Data Base.

Once it is created, it can be reviewed and changed at any time provided the status code is still in open (02) status.

Field names used on this screen are reserved words that are recognized by the PIREC application.

4.3. Telex Formator

A 'telex like' computer print out below was produced by the print program called from MPP114 (POTELEX ENTRY) by entering the action code "TR".

This print out is used for proof reading and hard copy approval to send out the telex.

```
AA  MOBIL PURCH + + +
ZCZC MOB001
RNYFP SINMO
.JAPTAMB FR 51696 PTALSM

PTA/PX/878912

/(BARLOW)(ROARK)(FERRANTE)/(ONG)(JEYA)/
EREQ PTAM71888
PRIORITY: ROUTINE
RUS: 25NOV89

ITEM: 1
QUANTITY: 2
UNIT: EACH
DESCRIPTION: SWITCH - DETECTOR. C/W EXPL. PROOF
             JUNCTION BOX.  MFG SMITH METER SYSTEMS.
             P/N 36-650381-375.
             PTA STOCK NO. 36-100-194. MES CODE, CCCN
             85.19.119
             ID 20PCT. VAT 10PCT. GATT PCT, STLG PCT.


COMCODE: 27650010

TOTALITEMS: 1
REQSUMRY: DETECTOR SWITCH
CTRYORIG: USA
MTHDSHIP: AIRFREIGHT
PFDVENDOR: PREVIOUS ORDER PTAM50346
           YOUNG MAN N CUFF INC

SUBACCEPT: NO
SPECINSTR: 1. UKPFROM SGS IS REQUIRED
           2. NOMINATED BANK DAGANG NEGARA
              CABANG LHOKSEUMAWE. ACEH UTARA,
              SUMATERA, INDONESIA
           3. NPWP 1.001.664.0-03
           4. CATEGORY 1
           ENDUSE - ARUN LNG PLANT - MAINT.DEPT
           5. PLEASE REQUEST RELATED SUPPLIER(S) TO
              SUBMIT PROFORMA INVOICE(S) EARLIER
              TO P.T. ARUN EXPEDITOR, ATTN,
              S.JEYAPATHY IN MOSPL SINGAPORE FOR
              RIB APPLICATION.
           RIZHAM/DJADJULI/ZAHRI
ESTVAL: 3500.00 USD
ENDREQ
RIZHAM/DJADJULI
 + + + +
140987/07:21WIB
```

4.4. Telex Interface

Upon approval of the hard copy, the operator enters his authorization code and selects action code TX to transmit the purchase order to its assigned foreign buyer.

The processes initiated by "TX" action code were:

1. Create formatted telex record on an index sequential file.
2. Call the dialogue program (PRTTLX) and the delayer (PAUSEPGM) program to establish a link with the DSX-40 message switch.
3. Transmit the index sequential file.
4. Disconnect the link.
5. When the file has been transmitted by the DSX-40, a telex copy is printed on the telex terminal installed at the user location.

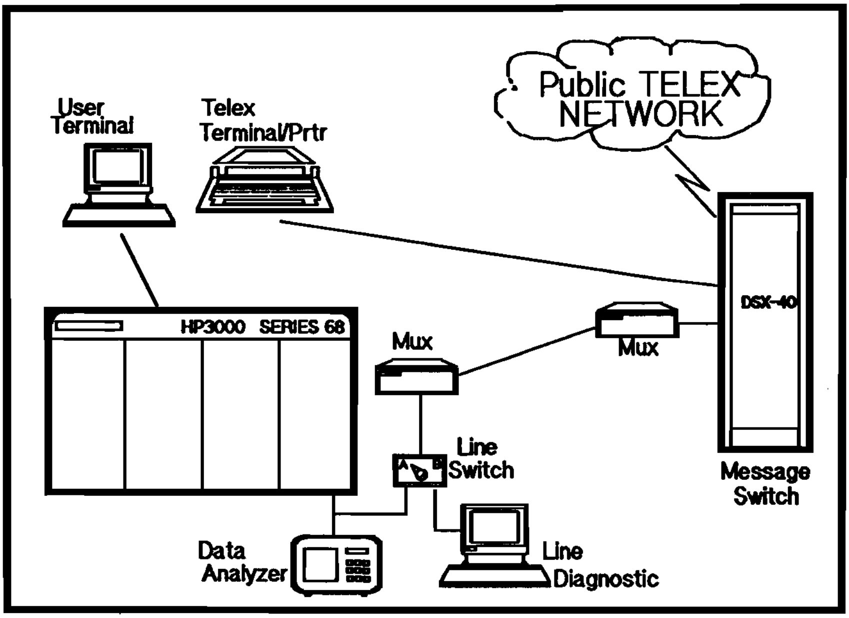
: ARUN. NGL. CO. TELEX P.T. ARUN. NGL. CO. TELEX P.T.

```
scsc 030106 381167
aa purch
++++
scsc mob001
rnyfp sinmo
.ieptamb fr 51696 ptalma

pta/pc/978912

/(barlow)(roark)(ferrante)/(ong)(jeya)/

ereq ptam71688
priority: routine
roe: 25nov87

item: 1
quantity: 3
unit: each
description: switch - detector, c/w expl. proof junction box.
             mfg smith meter systems, p/n 36-650381-375.
             pta stock no. 36-100-194,  mes code 67,  bm
             id 20 pct,  vat 10pct,  aid   pct,  etlg   pct.
coacode: 27550010

totalitems: 1
reqsumry: detector switch
ctryorig: usa
mthdship: airfreight
pfdvendor: previous order ptam50348
           young man n cuff inc
subaccept: no

specinstr: 1. lkp from ege is required
           2. nominated bank dagang negara
              cabang lhokseumawe, aceh utara,
              sumatera, indonesia
           3. npwp 1.001.884.0-03
           4. category 1
           enduse - arun lng plant - maint.dept
           5. please request related supplier(s) to submit
              proforma invoice(s) earlier to p.t.arun expe-
              tor, attn, e.jeyapathy in mospl singapore for
              rib application.
           risham/djadjuli/sahri
estval: 3500.00 usd

endreq

risham/djadjuli
++++
140987/07:21wib
```

TELEX P.T. ARUN. NGL. CO. TELEX P.T. ARUN. NGL. CO.

5. HARDWARE SPECIFICATIONS

Basic Hardware requirement used for this interface are HP3000 series 68 and DSX-40 Message Switching, connected via asynchronous port (ATP).



5.1. HP3000 Configuration

- Advanced terminal Processor (ATP) Port.

Asyncronous port as input & output to and from the HP3000 configured as follows:

Logical Device	= 56	
Device Ref Table #	= 8	
Unit number	= 36	
Channel number	= 0	
Type	= 16	
Sub type	= 0	
Terminal type	= 10	
Terminal speed	= 30	(300 bauds per second)
Record width	= 40	(in word; 80 characters)
Output Device	= 56	
Mode	= ID	(Interactive & Duplicative)
Driver name	= HIOTERM1	
Device Classes	= DSX40	

-HP232A terminal

One unit is located at the Purchasing department for maintaining purchase order data and sending the POTELEX. Another one installed in the computer room is used for line diagnostics and to check the connection to DSX-40 in case of transmission failure. Configured as follows:

TERMINAL Configuration

Terminal Id	= 2392A	Keyboard	= USASCII
Language	= ENGLISH	Local Echo	= ON
Capslock	= ON	Start Column	= 01
Bell	= ON	XmitFnctn(A)	= NO
SPOW(B)	= NO	InhEolWrp(C)	= NO
Line/Page	= LINE	InhHndShk(G)	= YES
Inh DC2(H)	= YES	Esc Xfer (N)	= NO
Fldseparator	= us	BLKTerminator	= rs
Return Def	= cr	Term Mode	= HP

DATACOMM Configuration

BaudRate	= 300	Parity/Databits	= ODD/7
Asterisk	= OFF	Chk Parity	= YES
SR(CH)	= LO	EnqAck	= NO
RecvPace	= NONE	CS(CB)Xmit	= NO
XmitPace	= NONE		

- HP1640B Serial Data Analyzer.

A device used to monitor the dialogue between the HP3000 and the DSX-40. Also used for monitoring the outgoing POTELEX.

*********** FORMAT **********	
DATA CODE	[ASCII-7]
MODE	[ASYNC-1]
BIT/SEC	[300]
ERROR CHECK	[ODD PAR]

*********** MODE ***********	
	[MONITOR]
TRIG SOURCE	[RX CHAR SEQ]
RUN MODE	[CONTINUOUS]
TRIG	[ ]
SUPPRESS	[OFF]

- Instamux-470 Multiplexor.

A pair of communication devices used to transmit POTELEX data over telephone lines to a DSX-40, located about one kilometer away from the HP3000 computer room.

- A-B Switch

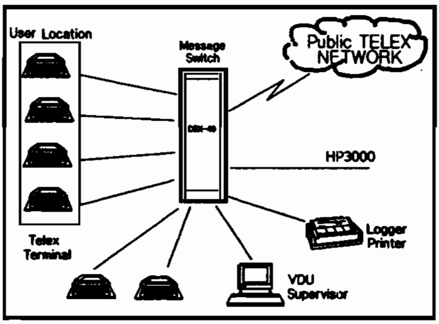
A device which is used to switch between two peripheral devices connected to an HP3000 port. Used in conjunction with line diagnostics.

5.2. DSX-40 Message Switching

The DSX-40 Message Switch is installed at another building separated about 1 kilometer away from the HP3000 computer room.

The configuration consist of:

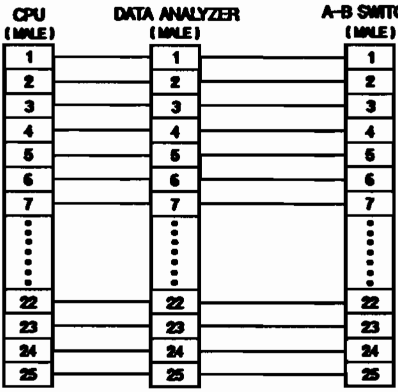
- One unit, VDU supervisor
- One unit, logger printer
- Six units, user station terminals, four of them located at the user site.

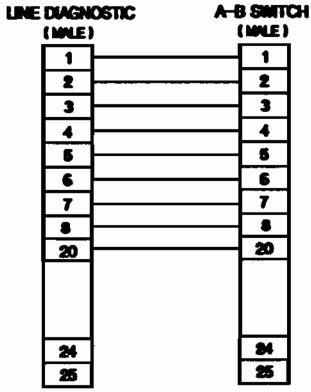


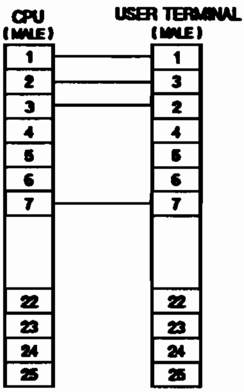
5.3. Physical Layer

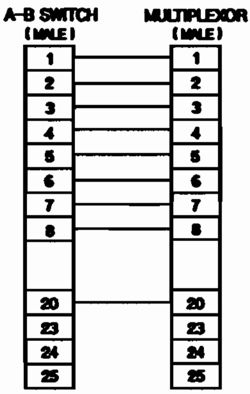
Connections between Data Communication Equipment (DCE) and Data Terminal Equipment (DTE) for this Interface generally use RS232 cables.

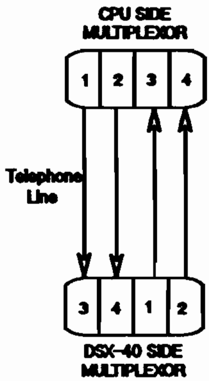
The pin connections for each piece of the equipment are drawn as follows:











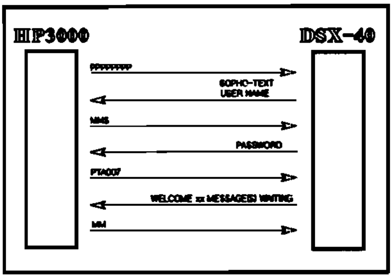
6. THE HP3000 - DSX40 INTERFACE

The main program for the HP3000 - DSX40 Interface is the MPP114 program. This program actually controls all the traffic between the HP3000 and the DSX40 such as:

- Add the telex addressee.
- Formatting the message in an index sequential file.
- Calling the dialogue program. (PRTTLX, PAUSE)
- Transmit the formatted message.

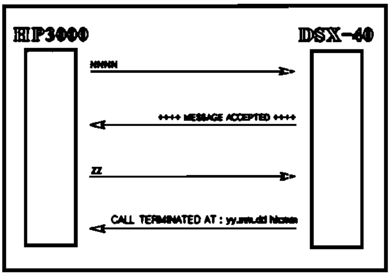
6.1. Dialogue

This process is initiated from the HP3000 program PRTTLX by sending a series of the character "P" at a certain interval until the DSX responds and prompts the user to enter their name and password.



After the line is established and the DSX40 responds with "nn MESSAGE(S) WAITING", the HP3000 sends the characters "MM" to DSX to indicate Ready To Send (RTS) data messages and start sending a telex formatted purchase order.

At end of the message text, characters "+ + + +" are added to indicate end of text. This will return the control to the PRTTLX program in order to send the characters "NNNN" to inform the DSX40 that the message has been completed.



6.2. Delayer (PAUSE program)

The delayer program, (PAUSEPGM) is a program that is written in System Programming Language (SPL).

This program acts as a "traffic light" that controls the outgoing and incoming message during the opening and closing dialogue.

The timer was set according to the time required by the DSX40 for each prompt.

7. Trouble Shooting

The interface program was prepared for user and operator convenience in case of transmission failure.

These failures occur if the HP3000 can not established the link to the DSX-40 after 50 times trial.

For the user, a certain period after he or she has entered "TX" action code, a message will be displayed at the bottom of the online screen (MPP114):

"1158 TRANSMISSION FAILED, PLEASE CALL EXT 2641 or 2671"

For the operator, a beep sound and the following message will be displayed on the console operator terminal:

```
PRTTLX FAILED
PLEASE CHECK DSX LINE
```

Obviously, before the failure occured the operator would know it, if he was watching the serial data analyzer (SDA) monitor. This device will display the dialogue from both the HP3000 and the DSX40.

Action should be taken following the transmission failure:

1. Position the A-B switch to connect to the diagnostic terminal HP2392A.
2. Open the dialogue directly from the key board.
3. If no response from the DSX-40, then check the physical connection.
4. When the line proves to be good, repeat step 2 until DSX-40 is responding and has terminated by itself.
5. Return the A-B switch to CPU position
6. Ask user to re-transmit the POTELEX.

8. COST AND BENEFIT

The expenses incurred for the POTELEX Interface program were two man months of an internal programmer/analyst salary. Data communications consultancy was also using internal man power assisted by a joint effort team from Mobil Oil Indonesia.

There was no additional cost for the devices required for the interface. All equipment installed was using existing equipment.

Benefits obtained by PT. Arun after the implementation of the POTELEX Interface were:

1. No duplication of work.

Data entered on the computer does not need to be retyped by the telex operator on the telex terminal.

2. Speed up material procurement processes.

Avoid delay of batch document to be processed manually by the buyer assigned.

3. Fast transmission.

At the speed rate of 300 bps, data were transferred electronically from the HP3000 to the message switching machine without human intervention.

4. Accuracy.

The contents of information received at the other end was similar, and without need of reformatting, as the outgoing file was already formatted to meet the input required by their system.

9. Acknowledgement

The authors would like to thank R.F. Hooper and J.J. Konz of Mobil Oil Indonesia for their technical assistance.

Also the authors would like to extend their thanks to P.T. Arun management, particularly Mr. R.L. Corkum, Mr. Oscar Tampubolon and Mr. J.E. Manopo for their effort to make this presentation possible.

HP, HP3000 are trade marks of Hewlett-Packard Company.

Instamux is trademark of MICOM Company.

DSX-40 is trademark of Philips Company.

A-B Switch is manufactured by RS Company.

PIREC, Purchasing Information and REquisition Control system is an application used by Mobil Corporate Purchasing, Fairfax Va.

10. BIBLIOGRAPHY

Interex Proceedings, Interex

DSX-40 Operating Manual, Philips

PIREC Systems user manual, Mobil Oil

HP3000 Systems Programming Language, Hewlett Packard

HP3000 Turbo Image Reference manual, Hewlett Packard

Booklet of PT. Arun Natural Gas Liquefaction Company

Do You Need A Fiber Optic Local Area Network?

J. W. Swearingen
The Apex Group, Inc.
7151 Columbia Gateway Drive
Suite F
Columbia, Maryland 21046

(301) 290-1606

Computer
Museum

7050-0

INTRODUCTION

Unless you have been living in a cave the last two years, you have probably been inundated with information on fiber optic local area networks. It is rare to pick up any issue of Computer World, PC Week, or most any other industry publication without seeing a significant discussion of fiber optics in its content.

This amount of attention on fiber optic technology is probably deserved. Never has a data distribution technology promised to change the way we process data more than fiber optics. Consider the following:

1) FDDI, a 100 Mbit per second (Mbps) fiber optic LAN standard is over 10 times faster than current popular ethernet networks.

2) Many manufactures have working 1 Gbit per second fiber LANs providing speeds 100 times faster than standard ethernet.

3) Fiber products are available today which allow over 1000 nodes to be connected over several kilometers without use of repeaters.

Clearly there are distribution options available today that are startling to say the least. For we HP users who have only had 10 Mbit per second IEEE 802.3 and ethernet for a few years, these options seem greatly beyond anything we are familiar with.

Given that network speeds are taking this leap forward, and products that contribute to or utilize higher speeds are finding wider use, now is a good time to examine the implications of "faster" networks and the technologies that surround them.

What I am going to try to accomplish today is to familiarize you with the key factors of fiber optic data distribution systems and why you should (or should not) think about implementing one. We will visit the popular standards, typical costs, advantages, disadvantages, and the future of fiber optics systems in our industry.

FIBER OPTIC LANS - OVERVIEW

Today, there are three popular fiber optic LAN implementations in use today. These are 1) 10 Mbit Fiber Optic Ethernet, 2) 16 Mbit Fiber Optic Token Ring, and 3) 100 Mbit FDDI or Fiber Distributed Data Interface. These products come from a variety of manufacturers and represent the most common implementations.

Token Ring Overview

The Token Ring Protocol, popularized by IBM and made standard by IEEE 802.5, allows the station holding the "token" to transmit a message at either 4 or 16 million bits/sec (Mb/s).

Stations are typically interconnected using a star-wired ring configuration which allows for easier fault isolation and maintenance. At each hub, there are Multi-Station Access Unit (MAUs) used to allow the insertion or removal of stations from the ring. Between 250 and 260 stations may be attached to a single ring, assuming that either IEEE 802.5 or IBM specifications are adhered to. Four conductor twisted pair copper (unshielded or shielded) or duplex fiber optic cables are used as media for station cabling. Generally, a distance of around 100 meters (for twisted pair) is assumed from a station to the MAU with an overall ring circumference that varies depending on the number of repeaters and the media types used (See Figure 1).

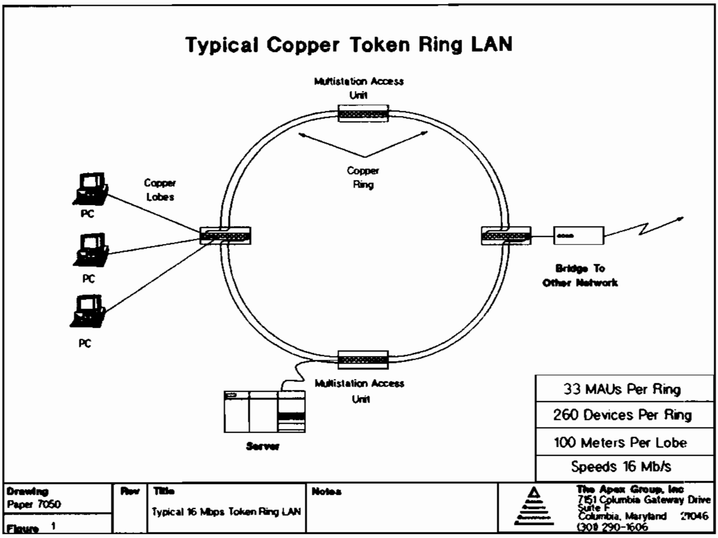


Figure 1 - Typical Copper Token Ring

The term "ring" is derived from the logical adjacency of the links between MAUs. That is, each connection has a physical link to an upstream neighbor and to a downstream neighbor. The term "token" refers to a special data sequence which is continuously sent around the ring. Any node which has data to send waits to receive the token. Only a station in possession of a token can transmit new data on the ring. Unlike Ethernet and other contention-oriented broadcast protocols, token passing resolves network access conflicts without collisions. This arrangement insures that every station on a token ring will always have access to the network within a predictable time interval, even under a heavy traffic load.

Fiber Optic Token Ring implementations can be made with fiber optic Token Ring Transceivers in a star/branching tree topology. That is, the distance between any given lobe and the next MAU or workstation can be up to 2.5 kilometers. These fiber optic implementations can be completed with standard 62.5/125 micron fiber optic cable. The use of fiber optic cable does not increase the speed beyond the standard 4 or 16 Mb/s (See Figure 2). On pure copper token ring networks, the use of fiber repeaters can provide distances of 2.5 kilometers between MAUs or fiber optic transceivers for extended lobes.

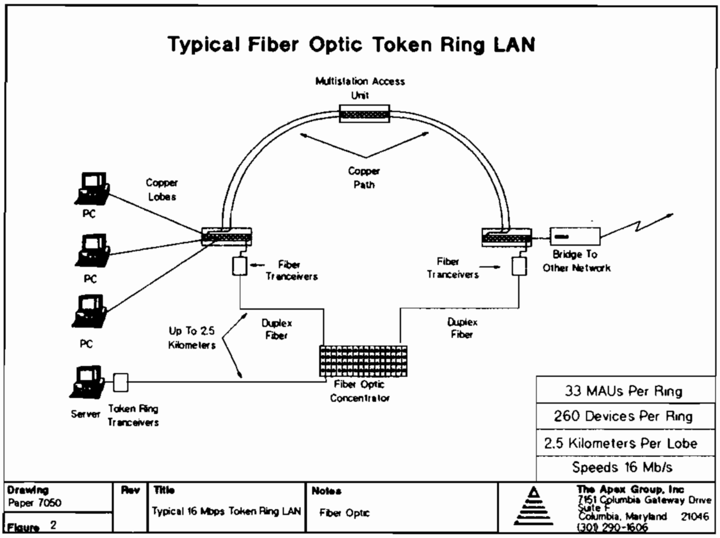


Figure 2 - Typical Fiber Optic Token Ring

Ethernet Overview

Ethernet is a high speed Local Area Network for exchanging data between information processing equipment within a moderate-sized geographic area. Ethernet is based on the CSMA/CD (Carrier Sense Multiple Access with Collision Detection) access method, in which a transmitting station first "listens" to the network to ensure that no other station is transmitting.

The specification for Ethernet is embodied in the IEEE 802.3 standard. Two baseband coax implementations exist, referred to as 10Base2 and 10Base5. These are commonly referred to as ThinLAN and ThickLAN respectively for the type of cable used and the distances allowed.

The 10Base2 implementation allows up to 30 stations along a 185 meter long thin 50 ohm coax bus. The 10Base5 version uses thicker, backbone cable where stations can be placed every 2.5 meters if desired. Each cable segment can be up to 500 meters long and can be interconnected using repeaters up to 2.5 km maximum station separation. Up to 100 stations may be connected to a single segment with a maximum of 1024 stations per network. Stations are attached to transceivers or Medium Attachment Units (MAUs) via an Attachment Unit Interface (AUI) cable. Data from stations are broadcast to other stations at a data rate of 10 million bits/sec (Mb/s) (See Figure 3).

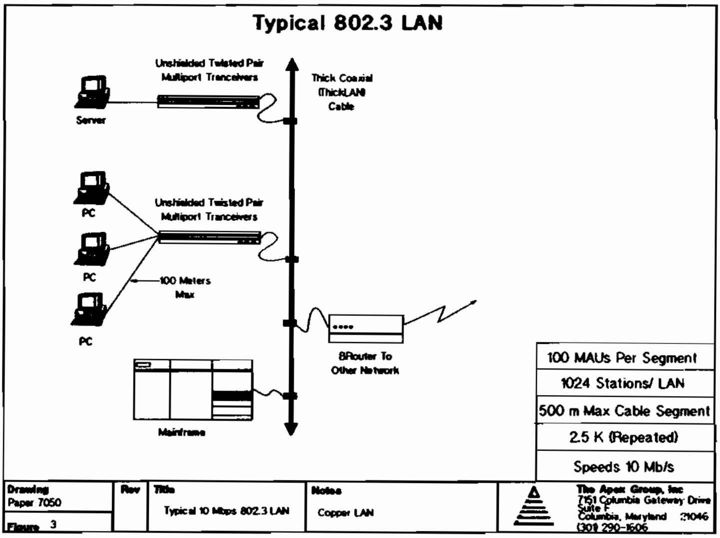


Figure 3 - Typical Copper Ethernet LAN

Fiber Optic Ethernet LANs are accomplished through the use of FOMAUs for Fiber Optic Medium Attachment Units and fiber optic concentrators (See Figure 4). These devices are typically used in a star configuration which allows for a high level of flexibility and the ability to be intermixed with existing copper (thick or thin) segments. These devices can also work in conjunction with network interface cards in workstations, multiport transceivers, multiport repeaters (Ethertwist), repeaters, bridges and other similar devices.

FOMAUs can be connected up to distances of 2.5 kilometers allowing for large scale LAN implementations. Figure 4 shows a typical LAN that is centered around a fiber optic concentrator. Individual fiber ethernet segments can be run from this concentrator for further data distribution using Ethertwist Hubs, host connections, wide area gateways or other compatible ethernet devices.

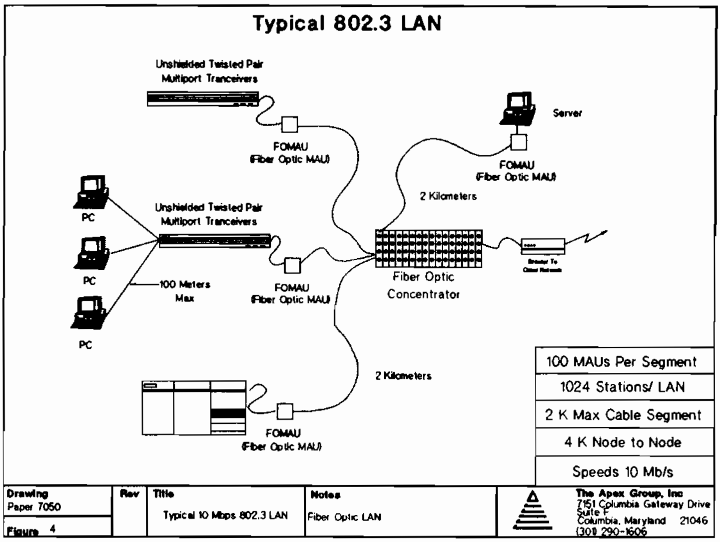


Figure 4 - Typical Fiber Optic Ethernet LAN

Many other manufacturers offer different physical packaging of these components but the basic technology behind the sheet metal is still basically the same. Companies like Synoptics, Cabletron, and Dupont for instance offer a card cage packaging allowing individual fiber concentrator, fiber and copper transceiver, and bridge cards to inserted as needed.

FDDI Overview

FDDI is a high speed access method very similar to Token Ring. It uses a bit pattern called a token to control media access. But FDDI uses two counter rotating rings. Data travels over one ring, while the second is used for a backup in the event of a failure. This "self healing" aspect provide extremely reliable networks since it can actually reconfigure itself, passing data over the backup ring and isolating the problem from the rest of the network. This type of redirection, however requires a double-attached station. Most stations that will be attached to FDDI will probably be single-attached due to cost, and practicality reasons.

While FDDI has officially not been completely ratified as a true standard, it is very clear that it will be a major force in the industry. Forrester Research Inc., a Cambridge, Mass. based market research firm, predicts that every major vendor will announce plans to support FDDI.

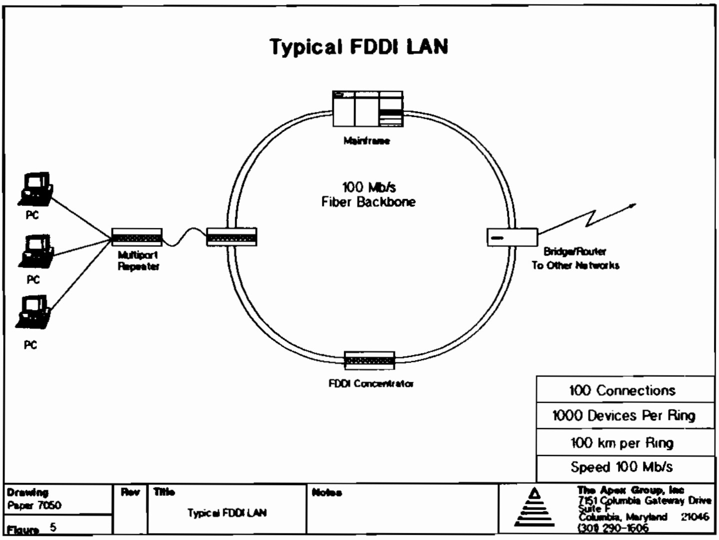


Figure 5 - Typical FDDI LAN

Figure 5 above illustrates a typical FDDI configuration. Todays products usually provide high speed backbones with slower ethernet subnets. These subnets can support multiport repeaters like IEEE 10BASET Ethertwist Hubs, host connections, wide area gateways and bridges. Some host based connections exist today allowing for a "native" FDDI connection directly to the host.

WHICH ONE IS RIGHT FOR YOU?

So, do you think you need one of these technologies? Well, most customers we speak to are not quite sure. If cornered, they usually say "I think so". The truth is, most people are fascinated with the technology of fiber optics but don't know if they need it now or in the future.

Why do organizations implement or think they should implement fiber optic systems? Stated another way, what features or benefits are usually mentioned by organizations contemplating fiber optics? The most frequent are:

1) **High Speed** - Fiber has the capacity of transmitting significantly higher speeds than traditional copper technologies.

2) **Noise Immunity** - Since fiber optics use light instead of electrical signaling, they tend to have better immunity to extraneous noise sources like EMI and RFI.

3) **Security** - It is a common belief that fiber optic cable cannot be "tapped" without notice thus providing better security features. Since there is no Radio Frequency (RF) emitted, wireless snooper devices cannot be used.

4) **Distance** - Fiber optic systems can transmit Local Area Network signals greater distances than copper systems.

5) **Ease of Upgrade** - While higher speed FDDI electronics are expensive today, prices will undoubtedly come down. Cable plants installed today can be upgraded to accommodate higher speed electronics.

6) **Electrically isolated environment** - This is similar to noise immunity and is the obvious light versus electric signalling benefit. This capability helps with electrical load balance between buildings in a campus environment.

These six reasons seem pretty powerful reasons for considering fiber optics. Things like corporate security, noise immunity, and high speeds are things all MIS or datacom managers would like to have. In fact, we think that the ultimate network is one that is the fastest, lowest cost, easiest to install, easiest to expand, and over all state-of-the-art. That's the theory anyway. We are all somewhat scared of being caught with our pants down and our existing networks become too slow and we are faced with the big "O" word (OBSOLETE).

It is important to identify each of your individual requirements and specifically evaluate fiber optics for meeting those requirements. Careful analysis could save your organization thousands of dollars today, and maybe more importantly, prevent you from jumping to soon into a technology that will have a very short usefulness.

ARE THE ADVANTAGES REALLY THERE?

Before your organization takes the big jump into fiber optics, it makes sense to reexamine the six so called features of fiber optics and make sure that the time is now, and the payoffs will *really* be there:

1) **High Speed** - This seems pretty obvious. Given the immense press coverage this has got to be one of the major selling points of fiber optic LANs, right?

   Not necessarily. There are two sides of this feature: 1) Do you need speed "relief" on your current LAN, and 2) Is the equipment you have or plan to have in the next few years, capable of supporting significantly higher speeds?

   Most copper ethernet or token ring networks in use today are using less than 10% of their capacity. These ethernet and token ring networks are yawning, begging for more traffic. We have seen multi-hundred node networks in very active environments never exceed 20% of capacity.

Microsoft corporation recently upgraded their corporate LAN to a FDDI backbone. The did this after finally experiencing performance and capacity problems in supporting 750 servers spread out into 14 buildings, connecting 6,000 personal computers, and lots of VAXs and AS/400s. This network is probably a far cry from what our environments resemble.

The second part of the speed issue is whether or not your computing and network electronics can keep up with these higher speeds. Getting network controllers, minicomputers, and workstations in today's organizations to utilize these speeds would be like asking you to drink from a fire hose. A typical IBM-AT class machine transmitting a file to another similar machine will, basically, not complete the task any faster over a 1, 10, or 100 Mbps backbone. The controllers and processors are simply not capable of anything faster.

Some test have been completed in the university environment where they eliminated most of the network layers and conducted straight memory-to-memory transfers. They used very fast workstations also. For instance a Sun 4 station completed file transfers at a very brisk 5 - 6 Mbps. This has to be close to the theoretical top end network performance that this Sun workstation can achieve and it is still very much short of anything approaching 100 Mbps. Effective network throughput is affected by many factors including network controllers, file systems used, and disk transfer rates.

HP Apollo, one of the few vendors that provides workstation that provide FDDI interface cards, can only effectively move data at speeds around 5 Mbps.

When you add the fact that Wide Area connections are not practically available beyond a few Mbps, and bridge technology is just now addressing the extremely computation intensive requirements of FDDI, we can really start to identify additional limiting bottlenecks that could occur over a high speed fiber network.

As we saw earlier, fiber optic LANs do not even transmit their signaling any faster unless you are using FDDI-like implementations. Fiber optic token ring and ethernet do not give you one ounce of extra speed.

Another interesting aspect of the copper versus fiber argument concerns the maximum speeds that will be attainable using standard twisted pair cabling. Many companies will be announcing 100 Mbps products using twisted pair cabling in the 1990 - 91 timeframe. This could greatly extend the life of existing fiber and potentially allow companies to postpone fiber expenditures.

2) **Noise Immunity** - This advantage is greatly overrated. We have never run into a office building that had an unmanageable amount of noise. Some factories are particularly bad - most are not. We recently proposed a fiber optic network to a manufacturer of industrial circuit breakers. They will routinely test these devices and the amount of

noise produced from these high voltage switches was incredible. In fact, the noise was so bad that CRT screen would sometimes literally explode. On the other hand, we recently recommended **against** fiber optics for a law firm in downtown New York when noise was their major justification for fiber.

Optical communications can have noise as well. Or the copper subnets that you install may be the weakest link in your noise vulnerability anyway. Unless you have an extremely hostile environment, fiber usually can not be justified using noise immunity as the primary reason.

Your environment can easily be tested to determine just how noisy it is and if common copper can relieve these sources.

3) **Security** - This is perhaps the biggest myth of fiber optic systems. In most organizations there are easier ways to snoop on a corporate network than tap into a copper backbone. Like walking into a wiring closet, plugging into any wallplate, or setting up a normal looking PC to log data to a disc file, modem, or floppy disc. If your backbone is fiber it does not protect any of these data access points.

   There are also techniques today that will allow a snooper to indeed tap into your fiber without cutting the cable. An if he wanted to cut the cable, he could do so, splice into a coupler, and reconnect in about 6 minutes. There are very few shops I know of (including some high security government customers) that could detect this intrusion - especially after normal working hours. Security is usually not improved by any significant degree with the implementation of fiber.

4) **Distance** - Fiber optic systems do indeed have superior distance capabilities. A Fiber optic repeater in an otherwise all copper backbone be a very good idea. This allows you to leverage off of you existing copper network investment without a total copper-to-fiber conversion. These devices can extend your LAN over 2.5 km in most cases.

   Good old copper repeaters used every 500 meters will solve most distance problems at a significantly lower price especially if you already have installed thick ethernet cabling. Sometimes getting to the cable at these short distance increments is difficult or the access points are not environmentally protected.

5) **Ease of Upgrade** - If you install fiber today, you will probably be positioned well to accept new fiber technologies in the future. Most cable installed today should be 62.5/125 micron size and be capable of 1300 nanometer wavelength.

   Jumping into non-FDDI fiber systems today in disguise as a ease-of-upgrade upgrade advantage, is only an advantage if you need fiber to begin with. Remember, copper isn't inferior or slow. Companies like AT&T and Ungerman Bass will be announcing products running 100 Mbps over common twisted pair. If copper networks will do, then don't use fiber. An if you are running copper, it may make sense to run fiber along with it and leave it "dark" for the future.

6) **Electrically isolated environment** - This is an advantage of fiber. However, proper installation of copper networks will prevent nearly all of the hazards associated with possible electrical disaster like lightening, high voltage shorting, and other such incidents.

   The key selling feature here is the ability to use fiber in environmentally harsh situations, especially where moisture could impair the performance of copper cable.

Of these 6 so called "reasons to buy fiber" only distance is a real reason today. There are some applications that could benefit from greater speed. These applications are not widely used today but will become increasing important to organizations over the next 5 - 10 years. They include:

1) **Image Processing** - Digital image processing includes the input, enhancement, analysis, and output of gray-scale or color-image data. These data are usually some form of "snapshot" of a real world object taken with a digital scanning device. Examples of these would be x-ray, Hubell telescope pictures of Neptune, or a map of Washington, DC.

   IBM has said that in the 1990s, electronic imaging will be its biggest growth area. The company has established a Department of Image Technologies with the IBM Research Division, and a publishing Systems Business Unit, chartered to printing markets. Hewlett-Packard has quitely made less noticeable but similar statements and today has a very strong product in the AIMS product family.

   Object oriented programming could mean that data bases could be built where single records are megabits or even gigabits in size. CD/ROM advances and multimedia capabilities on the desktop open up possibilities for desktop teleconferencing and extremely large file transfers. The files can range from 10 to 100 Mbytes each. These size files can increase the traffic on your network in a hurry making them a strong candidate for high speed fiber networks. Most industry sources believe that a super workstation will be required, capable of operating at speed of over 100 MIPS.

2) **Voice Annotation** - Similar in theory to imaging technology, voice annotation will provide digitized voice to accompany other non-voice files like spreadsheets, documents, images, etc. Special purpose chips from companies like Motorola will provide digital signal processing (DSP) chips at low costs and capable of 10 MIPS processing speed. This application too, will require large file transfer capability and large network capacities.

3) **HDTV** - One second's worth of bits required to fill an HDTV screen amounts to over one gigabit! Even with compression, FDDI will not even be fast enough.

4) **Others** - As organizations move closer to large client/server applications using a variety of compound objects, the ratio of LAN traffic to workstation compute cycles increase by orders of magnitude. More and more network capacity will be needed.

COST COMPARISONS

Let's use two examples that can illustrate some of the comparative cost of installing fiber optic systems. We will use examples of 1) a large office building located in a single site, and 2) a multi-building campus.

The first example of linking 20 floors of a large office building assumes an average of 20 feet per floor, two wiring closets per floor, requiring approximately 1000 feet of cable, and supporting 500 workstations and hosts. From the wiring closet to the individual wallplate would pretty much be a wash whatever method we used (Ethertwist, Cabletron, Synoptics, etc) so we will just consider backbone costs.

The straight ethernet coax alternative would cost the following:

Cable - 1000 ft. X $1.00/ft (Thick)	$1,000
Installation, mounting, testing	$2,000
ThickLAN transceiver (2/floor)	$5,000
2 ethernet bridges (filtering & distance)	$20,000
Total ThickLAN Backbone Cost	$28,000

A straight fiber optic ethernet solutions would cost:

Cable - 4000 ft. X $1.50/ft	$6,000
Installation, mounting, terminating,testing	$20,000
Fiber Optic MAUs and Concentrators	$100,000
2 ethernet bridges (filtering)	$20,000
Total Fiber Ethernet Backbone Cost	$146,000

Notice, this is a 421% increase without any real advantages over copper ethernet.

A FDDI backbone solutions would cost:

Cable - 4000 ft. X $1.50/ft	$6,000
Installation, mounting, terminating,testing	$20,000
FDDI Bridges 1 every 2 floors @ $30,000 ea.	$300,000
Additional subnet cabling (ethernet & copper)	$1,000
Total FDDI Backbone Cost	$327,500

This is a 124% increase over fiber ethernet and a 1,068% price increase over standard copper ethernet. The speed advantages of FDDI only an advantage if you need it. Distance is not a problem in this building and noise is probably manageable also.

A smart choice would be to run the coax and implement the standard ethernet topology with one difference. Also run the FDDI capable fiber along with it. The cable is relatively inexpensive, labor would only be increment by about 30%, and you will be ready for FDDI when the price comes down and you actually need it. If you do not think you will ever need fiber, do not run it.

A second example is three ethernet LANs over a one-mile span and can illustrate the cost of a campus environment. We will assume that the three buildings support 750 workstations and hosts residing on 3 discrete ethernet LANs. The linking of these three LANs will require around 6,000 feet of cable.

Assuming we can beat the 500 meter distance limitations of copper ethernet with repeaters or bridges, the following cost comparisons would apply:

Straight ethernet coax solution would cost the following:

Cable - 6000 ft. X $1.00/ft	$6,000
Installation, mounting, testing	$8,000
3 ethernet bridges, repeater, fusing	$31,000
Total ThickLAN Backbone Cost	$45,000

Straight fiber optic ethernet solutions would cost:

Cable - 6000 ft. X $1.50/ft	$9,000
Installation, mounting, terminating,testing	$25,000
Fiber Optic MAUs and Concentrators	$20,000
3 ethernet bridges (filtering)	$27,000
Total Fiber Ethernet Backbone Cost	$81,000

Notice, this is a 80% increase over copper ethernet.

A FDDI backbone solutions would cost:

Cable - 6000 ft. X $1.50/ft	$9,000
Installation, mounting, terminating,testing	$25,000
FDDI Bridges - 3 @ $30,000 ea.	$90,000
Total FDDI Backbone Cost	$124,000

This is only a 53% increase over fiber ethernet and a 176% price increase over standard copper ethernet. Since no real speed advantage can be realized using the fiber ethernet implementation over copper ethernet, either the copper ethernet or the FDDI solution should be implemented. Again, the speed advantages of FDDI is only an advantage if you need it. But with this many workstations being supported, the extra cost involved with FDDI is probably justified.

In five years FDDI hardware is expected to cost about the same as ethernet hardware. Thus, by installing fiber optic cable now and waiting for prices of FDDI to drop, a network manager can move to a theoretical tenfold increase in bandwidth for the cost of replacing the ethernet electronics. This will be very important if this company chooses to upgrade their subnet electronics to FDDI.

A network manager or MIS Director that chooses to solve his problem using just coax cable would save money today, but would face much higher costs to upgrade the network to FDDI in the future. In five years, you would be faced with recabling and adding FDDI hardware. The price of cable will drop slightly; the cost of installation will undoubtedly have increased 6% per year (30% over 5 years). This will bring the installation cost to around $32,500 in addition to the $14,000 to have the coax cable installed in the first place.

CONCLUSION

What's the bottom line? It really does depend on your organization's specific requirements and physical plant. The most important part of this discussion is that you should not automatically assume you need a fiber optic LAN. Many of the commonly believed benefits of fiber optics in a LAN environment such as higher speeds, better security, noise immunity, and greater distances may not be an advantage to you at all. Additional expenditures for fiber optic systems may not be justified if standard copper LANs will accomplish the same thing.

The emerging FDDI standard could change the way we process data. However, most computer resources in use today can probably not take advantage of the increased speeds. As FDDI component prices drop, increased workstation and mainframe performance proliferates, and high network-use applications like imaging and HDTV emerge, FDDI (or faster) will be required. FDDI can also supply relief to very large scale networks when acting as a backbone technology.

Figure 6 below illustrates the 5 technologies discussed in this presentation. MIS and Network managements must ask themselves which network technologies will position their organization for the future while meeting todays requirements at the lowest cost.

LAN COMPARISIONS - COPPER vs. FIBER

Technology	Max Distance	Max Distance W/Repeaters	Physical Connections	Speed	Noise Immunity	Security	Cost
Copper Ethernet	500 METERS	2.5 KM	100 MAUs 1024 STATIONS	10 MBPS	GOOD	FAIR	LOW
Fiber Ethernet	2.0 KM	4.0 KM	100 MAUs 1024 STATIONS	10 MBPS	EXCELLENT	FAIR	MODERATE
Copper Token Ring	NODE DEPENDENT (-500 METERS)	2.5 KM	260 DEVICES 32 MAUs	16 MBPS	GOOD	FAIR	LOW
Fiber Token Ring	2.5 KM	5.0 KM	260 DEVICES 32 MAUs	16 MBPS	EXCELLENT	FAIR	MODERATE
FDDI	100 KM	100 KM	100 CONNECTIONS 1000 DEVICES	100 MBPS	EXCELLENT	FAIR	HIGH

Figure 6 - LAN Technology Comparison

While we did not spend much time talking about fiber to the desktop, it is nearly unanimous in the industry that advances in desktop processing speed, and reduction in FDDI components must be present before it becomes practical or cost effective. As of this writing, less than a half dozen vendors supply FDDI network interface cards. The prices for these cards range in price from $8,000 - $12,000 each. Industry estimates calculate this cost reduction period to occur within the 4 - 6 year timeframe

Fiber Optic LAN speeds are also sometimes misleading, and may not be a good characterization of actual performance. Many parameters can determine speed including operating system, disc speed, I/O architecture, network controllers, media to the desktop, and more.

The final word is, be prepared for a fury of new information to keep being thrown at you. Things are guaranteed to change at an unprecedented pace relative to this topic. When trying to choose technology for your organization's network however, spend 80% of the time you have allotted for this decision on **understanding your organization's requirements** - now and five years from now. Research the industry, interview the key strategist within your company, and find out how organizations in your business will be processing data in the 90's. The time spent on those activities will be a much better use of time than understanding

the latest network technologies and probably will change at a much slower rate. Besides, as you have probably found out by now, staying on top of the latest network technology is nearly impossible anyway!

BIBLIOGRAPHY

Fahey, Michael. "FDDI is here but users will wait on fiber-to-the-desk." *Lightwave*, April 1990, p. 6.

Frank, Howard. "Managing networks versus networking management." *Networking Management* , December 1989, pp. 62-63.

Hancock, Bill. "Put More FIBER In Your Network." *DEC Professional*, October 1989, pp. 36-47.

Harbert, Tammi. "Image-processing vendors gear up for fast growth." *Electronic Business*, pp. 72-74.

Heywood, Peter, and Greenfield, David. "FDDI Just Say Not Yet, High Costs, Looming Interoperability Problems, and Lack of Bridge Standard Should Be Considered." *Data Communications*, April 1990, pp. 45-52.

Jenkins, Avery L. "Increased Backbone Traffic Pushes Users Toward FDDI." *Digital Review*, February 5, 1990, pp. 39-43.

Partridge, C. "A Faster Data Delivery." *UNIX Review*, March 1990, pp. 43-48.

Special Report FDDI. "Vendors focus on throughput limitations." *Lightwave*, February 1990, p. 1

TITLE: Experience with Distributed X.25 Multivendor Networking

AUTHOR: Alan MacIntosh
Hewlett-Packard
c/o Ella Washington
19091 Pruneridge Ave. MS 46LK
Cupertino, CA 95014

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 7055

TITLE: Experience with Networking-Moving to x.25

AUTHOR: John Howitt
ECC Building Products
Hulland Ward
Derby Derbyshire DE63ET
England

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 7056

Growing Your LAN into a WAN-LAN

Michael Russell
Hewlett-Packard Company
P.O. Box 152030
Irving, Texas 75015

Technology has allowed computer network designers to redefine the "local" in Local Area Networks. This paper traces techniques that allow LANs to be extended beyond the local environment of a single building to a multi-building campus network or even a country-wide network.

Introduction

As a common point of reference, definitions may be helpful. The Local Area Network (LAN) came into existance several years ago. It provided an efficient means of communication for distributed computer systems in a local setting.

The Wide Area Network (WAN), on the other hand, existed for many years prior to this. It dates back to the early Remote Job Entry systems for mainframe computers. The Wide Area Network typically involved systems located remote to one another, sometimes across many miles. Dialup or leased communication lines provided the connection for data transfer.

In the discussion to follow, the application of wide-area techniques to the concept of LANs will show how a LAN can grow to become a WAN-LAN.

History of LANs

What is "local" anyway?

In the beginning, the term "local" implied systems located within a workgroup or department, or definitely limited to a building. Telephone communications were not used; hardwired connections were the rule. This presented problems if the need existed to talk outside the building. LANs, however, did provide higher speed transmission and supported much greater distances than RS-232 connections.

The PC Influence

While computer literates debated the pros and cons of using LANs, an explosion in the industry solidified their acceptance. The personal computer came into being! Users requested serial connections to minis in order to support terminal access and occasional file transfer.

Companies such as 3COM and Novell were founded, helping to create the market for PC networks based on LAN technology. PC users recognized there could be benefits in using servers for central file storage and sharing of files, laser printers, and other peripherals.

Topologies

Several types of LAN network topologies developed and continue to be used today. With the bus topology, computers are connected in a multi-drop fashion along the length of the LAN cable, as shown in Figure 1. The LAN is terminated at each end and has taps for each node on the network. This scheme lends itself to backbone wiring which will be discussed in more detail later.

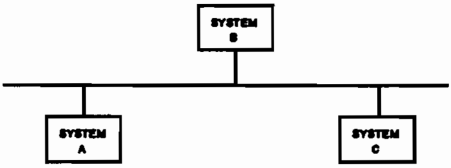


Figure 1

The ring topology is a complete loop of cable with devices dropped along the way. One disadvantage this presents is that the LAN cable must continue from the last node back to the first on the network. Figure 2 illustrates the ring.

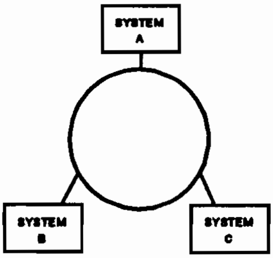


Figure 2

Another common topology is the star. It provides a central connection device with LAN cabling branching out to each node as shown in Figure 3. It is analagous to telephone technology with a central exchange, with individual wiring to each telephone from that point.

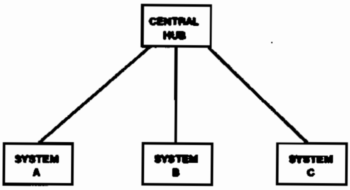


Figure 3

Standards

The evolution of standards is one of the factors that has helped the proliferation of LANs. Ethernet was an early defacto standard that contributed to the IEEE 802.3 standard. While these terms are sometimes used interchangably, they are different. They do agree from the hardware perspective, with slightly different data packet formats.

3COM was and is an advocate of Ethernet for its PC LAN connections. Hewlett-Packard has followed the IEEE 802.3 standard. IBM has chosen the IEEE 802.5 route for its Token Ring LAN. These standards refer to the physical and data link layers of the OSI model, as shown below:

7	Application	Service for application
6	Presentation	Data formatting
5	Session	Logon/other service facilities
4	Transport	End-to-end transmission check
3	Network	Information switching/routing
2	Link	Transfer of data
1	Physical	Transmission to physical medium

Software for the transport layer allows for end to end data transmission. Novell has its own proprietary transport, but its network software runs on different types of LAN connections, such as IEEE 802.3 and IEEE 802.5.

Most vendors of network equipment today who support Ethernet also endorse the 802.3 standard. Networks of different types can often communicate through the use of bridges or gateways. For example, Hewlett-Packard supports a LAN gateway for connecting a Novell network to its OfficeShare/Lan Manager network.

The important thing for LAN implementors to remember is to choose vendors who comply with standards. As their network grows to an organization-wide environment, expansion comes easier and the network investment is protected.

LAN Wiring

There are primarily three different types of wiring used for Local Area Networks:

(1) Coaxial cable
(2) Twisted pair
(3) Fiber optic cable

Coax cable comes in a variety of types and sizes itself. ThickLAN is the classic Ethernet 10 mm diameter 50 ohm coax. Because segments can be up to 500 meters in length (3 times that by using repeaters), it is an effective backbone cable for LANs. Due to cost and relative difficulty in bending, ThickLAN does not make a good cable for individual workstations.

ThinLAN or "cheaperNet" as it has been called is more appropriate for connecting PCs to a network. This coax supports the same transmission speed of 10 Mbps as ThickLAN and is only 4.9 mm in diameter. This type of cable looks similar to cable television(CATV) cable but is 50 ohm RG-58 as opposed to 75 ohm RG-59. Early PC LANs of bus topology used ThinLAN.

Unshielded twisted pair cable initially supported speeds of only 1 Mbps. However, with the advent of Hewlett-Packard's StarLAN 10, transmission speeds of 10 Mbps were achieved, as with the coax discussed earlier. (Twisted pair also is available with shielding as required by IBM's Token Ring network.) Today, the media of choice for connecting workstations is unshielded twisted pair. It not only is flexible to make installation easier but also is inexpensive and may exist currently (if extra telephone pairs were installed).

Fiber optic cable is a newer technology for use in LANs. This media offers distinct advantages where longer distances and immunity to electromagnetic interference are required. Fiber provides an excellent solution for installations where separate buildings must be connected. The 10 Mbps speed of coax and unshielded twisted pair does not even challenge fiber optic cable's bandwidth. For this reason, LANs will see even wider use of fiber optics in the future.

Let's turn the discussion of LAN wiring from the type of media used to the environment to be served. Many workstation networks in place today are standalone workgroup or departmental "sub-networks." They exist to provide file and print resources for a group of PC users. The media used may be a variety of the types discussed earlier. What is required to allow these users to share information with other departments? A facility backbone is needed. The backbone provides a means of connecting subnets and is typically implemented using coax or fiber. As discussed earlier, ThickLAN is a good choice for backbones. Not discussed was the fact that this media provides "baseband" transmission of data. Another type of technology, "broadband" also uses coax and supports the bandwidth to carry multiple data channels, video, and voice. This is the same technology used for cable TV. While this is a more expensive alternative, broadband may provide a good technology for applications within a manufacturing operation. Ungermann-Bass markets buffered repeaters to bridge StarLan 10 twisted pair subnetworks. Figure 4 shows an example of subnetworks off a building backbone.

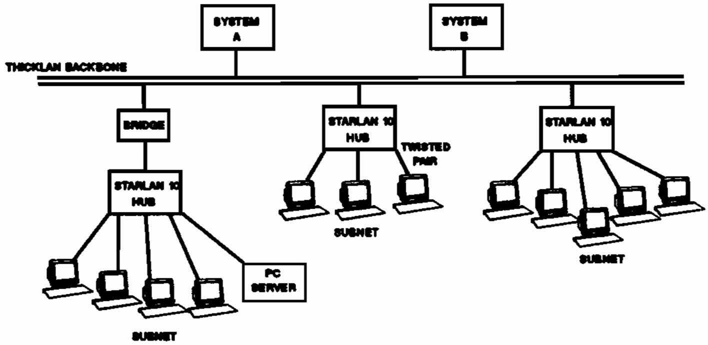


Figure 4

Notice that this network also includes a bridge to isolate traffic in the PC server subnet, thereby improving the performance of the overall network.

Advantages of LANs

Let's go back in time again to the early days of LAN acceptance and consider how System-to-System networks were implemented. How did we network three HP3000s that all needed direct access to each other? (The assumption is made that they were in the same building.) Figure 5 shows that a total of three network cables and two interface links per system are required. Also shown in this figure are the requirements if one system is added to the network. Six cables with three interface links per system are needed. Hardware costs rise quickly as systems are added.

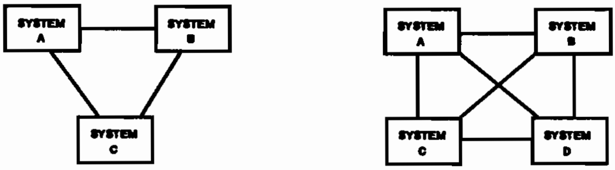


Figure 5

Applying the technology of LANs to this problem yields the results of Figure 6. No matter how many systems are added, there is only one interface link per system and one backbone cable with a tap for each system.

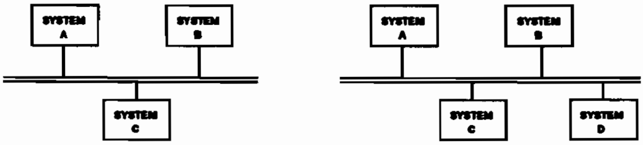


Figure 6

This is a primary reason why LAN technology lends itself so well to the networking of PCs. One connection to the LAN from a workstation gives access to any server resources available. Resources such as discs, printers, plotters, and modems are shared.

High speed data transmission is another advantage of LANs. This factor has been important as the growing number of personal computers has resulted in a need to integrate them into companies' information systems strategies. CPU intensive applications such as word processing and graphics have benefitted from the move to PC based software. Virtual terminal access to mini or mainframe applications such as data base and electronic mail have demanded the integration of PCs with larger systems. The subnet concept shown earlier is further defined with application sources shown in Figure 7. The individual user has a multitude of resources available through the use of the Local Area Network.

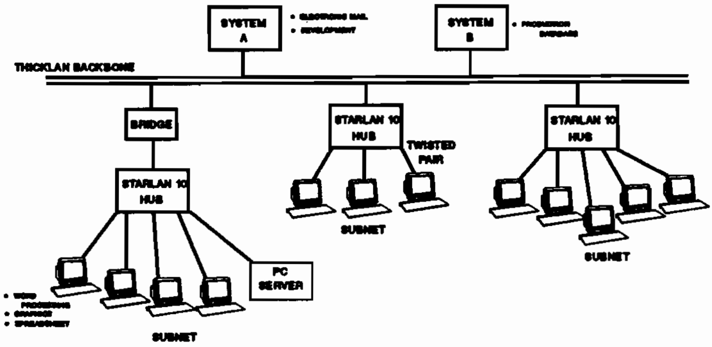


Figure 7

Case Studies

This section presents three actual customer situations where information needs were met through the use of LANs. In each case the following outline is covered:

(1) Environment
(2) Requirements
(3) Network Design
(4) Results

Law Firm

Five years ago a large law firm located in a major metropolitan area decided to move to PC based word processing for their two hundred fifty attorneys. Their facility was located in a multi-story building in the downtown area. The senior partner who internally sold the project for office automation was also committed to electronic mail as a productivity improvement. The firm wanted business software to do their accounting functions as well.

They chose Hewlett-Packard as the vendor to support these needs. Initially, the word processing was implemented using TouchScreen II personal computers (the Vectra didn't exist yet, remember). The extent of the PC network was two PCs sharing a LaserJet printer; they were not connected to the HP3000 which ran third party legal accounting software. The standalone PC pairs were installed throughout the ten floors of the building with a small cluster of terminal users on one floor in the accounting department.

The second phase of the project was to provide shared file resources for the word processing (PC) users as well as firm wide electronic mail. This need was to be met using the HP3000 with Resource Sharing and HPDeskmanager. Figure 8 shows the network design that was implemented.

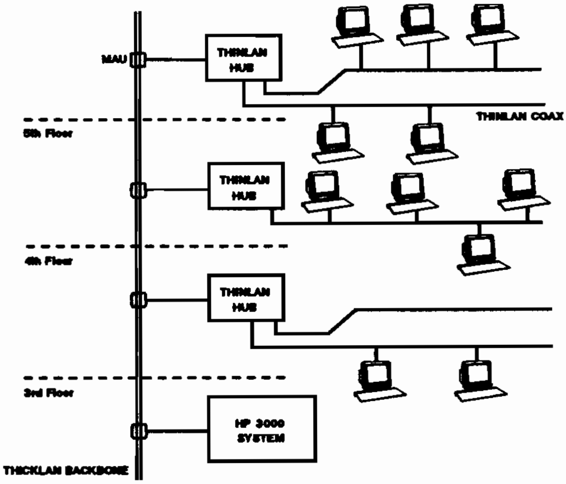


Figure 8

Since StarLAN 10 did not yet exist, ThinLAN was chosen as the media to connect the workstations. A backbone ThickLAN was run vertically from floor to floor with ThinLAN Hubs on each floor.

The network provided attorneys' secretaries with the ability to share "boilerplate" documents and contracts without using floppy discs and "WalkNet." It also gave users throughout the building terminal access to the electronic mail facility on the HP3000.

College Campus

Two years ago a college in northern Texas began examining their needs for integrating personal computers with their administrative computer center. This site had for several years been using HP3000 computers for registration, accounting, word processing, and other administrative applications as well as academic use in its computer science department. Isolated islands of personal computers were springing up in various departments around the campus. A few PCs were connected as terminals via serial links to the HP3000. HPWord was being used as the host based word processing solution. Users in the on-campus TV station wanted to implement a multi-user PC based software package for managing their telethon donations. The decision to move to PC based word processing via WordPerfect was made. Electronic mail was also a desired feature.

Although a LAN based solution seemed approriate, the problem of what type of wiring to use in a multiple building environment had to be addressed. Fiber optic cabling answered the need, but alternatives existed in the design of the network. With the administation building housing the HP3000, there were initially only two remote buildings that needed access. This requirement could have been met as shown in Figure 9 by using back to back Siecor company EOT 322 Fiber Optic Transceivers. Other buildings to be added to the network later would require the same arrangement.

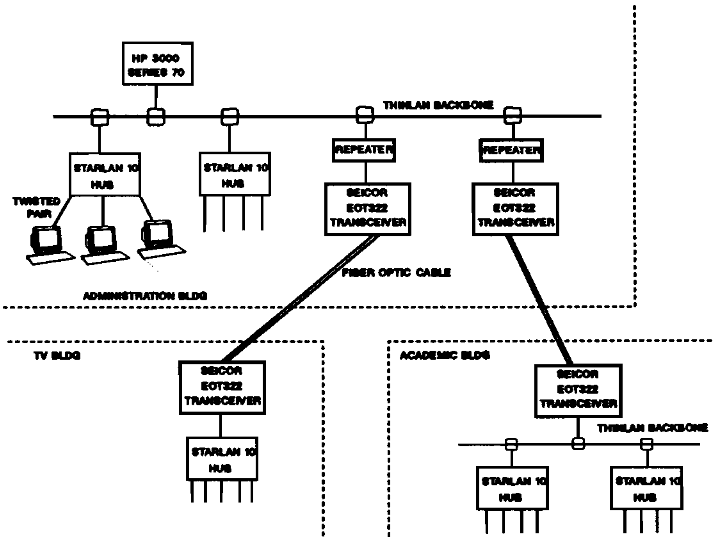


Figure 9

The chosen alternative provided more flexibility for slightly higher cost. By using the Siecor Active Star Concentrator, remote buildings were connected with the fiber optic cable and a single transceiver at the remote end as Figure 10 shows. This Star Concentrator supports up to 32 connections.

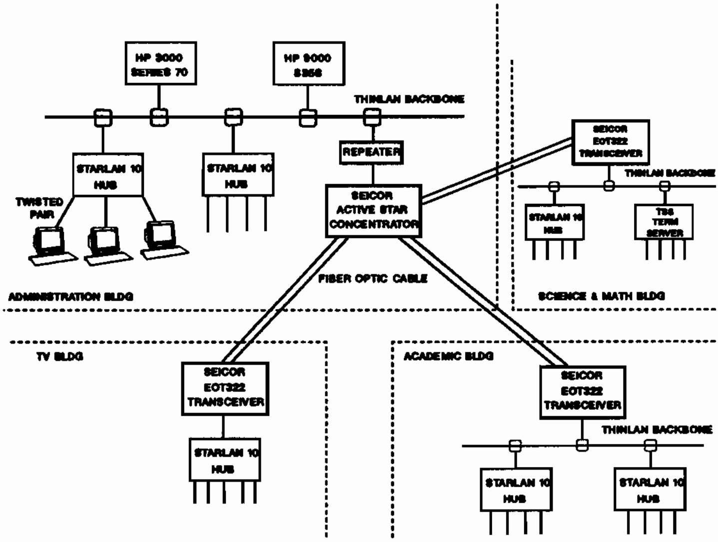


Figure 10

Support for PCs is through the unshielded twisted pair of StarLAN 10. The StarLAN 10 Hubs in the administrative building are dropped off a ThinLAN backbone in the HP3000 computer room. StarLAN 10 Hubs in the remote buildings are connected directly to the Siecor transceivers if 12 or fewer ports are required. Otherwise, a ThinLAN segment in the remote building supports multiple StarLAN 10 Hubs.

Since the network was initially implemented, an HP9000 has been added to the system backbone to support science and math users in a new remote building.

Gas Marketing Company

More than three years ago a natural gas marketing company headquartered in the Dallas area undertook a large automation task. The firm had offices located in six cities across the U.S. Their business involved marketing gas to large corporations, cities, universities, and other clients. There was a need to use Lotus spreadsheets. PC based word processing and graphics were also required.

The company chose the HP3000 with TurboImage to develop its gas marketing system. Each of the larger cities was to have its own local HP3000 with a database containing gas availability for that region. Smaller offices could be connected via network to the headquarters system. Use of HP Vectra personal computers was extensive.

Classic system to system Wide Area Network technology would have employed leased line or X.25 links. Shared resources for the PC users was a primary design consideration, however. During implementation of the headquarters system, the customer realized the power of allowing users to share spreadsheet files and print resources. They opted for departmental LaserJet 2000 printers. The question arose of how to allow remote site users access to local user resources and vice versa. In other words, local users wanted the ability to easily retrieve a remote spreadsheet or print a PC document on a remote printer. They wanted to use the advantages of the LAN on a wide area basis, hence the term WAN-LAN. The resulting network design is shown in Figure 11.

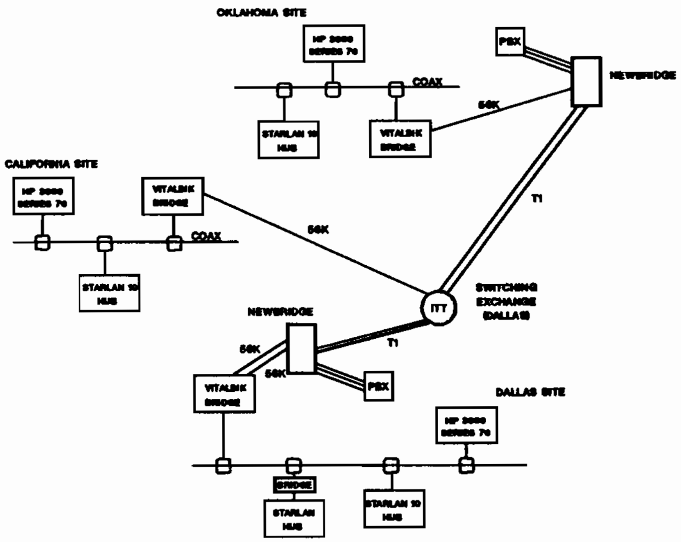


Figure 11

The Vitalink TransLAN bridges at Dallas and Oklahoma City were connected to T1 lines via Newbridge T1 channel bank equipment. This allowed voice and data transmission over the link. The data lines split from the T1 were 56KB. A direct 56KB link was implemented to the California site using the Vitalink bridge. Initially media used at the Dallas site was StarLAN, the 1Mbps version of unshielded twisted pair. Later portions of the network used StarLAN 10. The net effect was that this network appeared to be a single Local Area Network. A single user had the following capabilities with the WAN-LAN: printer access to any printer on the WAN, plotter access to any plotter on the WAN, private and shared disc resources anywhere on the WAN (assuming correct passwords), terminal access to electronic mail and gas marketing applications to any HP3000 on the WAN.

Future Directions

Network Management

As networks become more complex and span larger geographies, companies face a need for effective network management. This is the ability to monitor as well as control and diagnose the network. The International Standards Organization (ISO) has identified five Network Management functions:

Fault Management	- Ability to identify, diagnose, and resolve network problems
Performance Management	- Allows network operators to collect and analyze data about network in order to optimize performance
Accounting Management	- Provides information on use of network to allow future growth
Configuration Management	- Allows the network manager to track configuration and change
Security Management	- Ensures that only the correct people gain access to data over the network

Hewlett-Packard's OpenView Network Management strategy also adds the category of networked system management. This gives network managers the ability to manage networks from a central point.

While network management is recognized as a key factor, standards are still evolving. Most new network products are being introduced with some type of network management capability. As an example, now that the 10Base T standard of unshielded twisted pair has been adopted, HP has introduced the EtherTwist Hub as an upgrade for StarLAN 10 Hubs.

Standard Network Operating Software

The use of Lan Manager for OS/2 and UNIX environments provides an industry standard server for network resources. Higher performance and increased flexibility are characteristic of this software. Microsoft, 3COM, and HP have all worked together on the development of these products.

Cooperative Computing

Now that networks have evolved to the point of almost endless resources for the PC user, another industry buzzword has appeared, "cooperative computing." Servers are becoming more transparent to the user while at the same time becoming more prevalent. In the past PC applications ran on the local user's machine while large database or other specialized applications ran on the mini or mainframe. The distinction between PC and mini based software is blurring. New applications are being introduced which have portions of their software running on the PC with hooks into software running on the mini. One example is HP's NewWave Office Information Access software, a database inquiry tool. Hence the integration of personal computers and servers/minis is becoming tighter and tighter.

Conclusions

Hopefully, this discussion has helped clarify where we have come from in the realm of Local Area Networks. This technology has become more and more important with the integration of personal computers and larger systems. Certainly the term "local" has taken on a new definition as the ability to span not just building floors but buildings and cities has become a reality. And yet, new capabilities and standards continue to be developed. Where do we go from here?

Brice Clark and Charles Breed
Hewlett-Packard
8000 Foothills Blvd.
Roseville, CA 95678

INTRODUCTION

Most business people would agree that ready access to information is critical to a competitive enterprise in today's fast-paced business environment. It is also increasingly clear that businesses must operate in the global marketplace. Today's corporations are either expanding into new markets or being threatened by expanding global competitors. Together, these two factors are driving us into the age of the global LAN.

Business information technology in the 1990's will use high-speed networks that span the globe. Information will flow transparently between mainframe, midrange, and personal workstation systems as if each were at the next desk. This unprecedented connectivity will be the cornerstone of the global enterprise.

This paper discusses the networking transport infrastructure for enterprise LANs by addressing the following aspects:

* The changing nature of computing
* Transport needs of the new computing environment
* Key selection criteria and functional components of a LAN

The Shifting Computing Environment

While a global LAN seems like a contradiction of terms it is a natural result of the proliferation of LANs at the department and site level. We are really talking about the interconnection of workgroup LANs across a site and site LANs across wide-ranging geographies -- the global , or since we are taking about businesses, the enterprise LAN.

The rapid deployment of PC LANs and "distributed-processing" is a key force pushing us toward an enterprise LAN, as opposed to other network alternatives. Initially, PC LANs were installed to allow the sharing of storage and peripherals. Information access to a mini/mainframe data center or corporate system has usually been accomplished by terminal emulation at the personal workstation.

Now, however, the personal workstation LANs -- by which we mean LANs of PCs, Macs and UNIX workstations -- can be viewed as a valid and highly flexible computing platform. This new platform, often called client-server computing, represents a major new computing paradigm that challenges conventional thinking about applications and the mini/mainframe environment.

The attractiveness of this new platform is in part due to the existing installed base of personal workstations and departmental LANs. In other words, many businesses find they already own most of the hardware infrastructure (at least the PC's if not the LANs) for the new environment. The second factor is the power of the personal workstation to deliver easy-to-use, highly integrated, modular applications when coupled with powerful servers performing as distributed data repositories. This second point holds the key to how companies -- and the industry -- will move from the old environment to the new. Today, most data exists on minicomputers and mainframes. It is far more desirable to leave the data there than to move it. As is often the case, the answer will be to integrate the "old" with the "new" -- minis and mainframes will be "connected" to the client-server platform primarily as servers.

Transport Needs of the New Computing Environment

It is clear that the new computing environment is LAN-based. As this new LAN-based platform becomes increasingly deployed, the natural evolution will be the interconnection of departmental platforms -- actually department LANs -- via facility-wide backbones. This is similar to the way mainframes and minis -- the principal applications platforms -- evolved into networked systems as the need to share information between applications grew. However, unlike minis and mainframes which when connecting between geographically remote sites used low-bandwidth point-to-point links or packet-switched networks, today's lower transmission costs and higher available bandwidth allow a different and far more powerful approach -- extended LANs. Coupling the lower costs for bandwidth with robust BRIDGES and internetworking ROUTERS allows the creation of high-speed corporate backbones linking major facility networks into a single high-speed LAN.

With a LAN-based application platform, the transport system must be very robust. It must offer high performance because information that once stayed within a single machine will now be moving over the LAN between clients and servers. The new LAN transport should, as much as possible, be protocol-transparent because the new environment will not be made up of just one vendor's equipment. In fact, much of the equipment is already in place and it has been purchased from many vendors already.

It is the way this equipment will be used that is new, not the equipment -- at least for the most part. If one can put in place an infrastructure that essentially behaves as a single LAN -- including performance characteristics -- then the new-breed applications can actually operate across multiple departments throughout a geographical dispersed enterprise as if everything was located in the same room.

Key Elements for Creating the Enterprise LAN

To qualify as an enterprise LAN supplier, a vendor must offer integrated solutions that move the LAN from an ad hoc, locally administered phenomenon into a true enterprise-wide, data communications infrastructure. Our customers have described their requirements as follows:

Functionality and Performance: The resulting network must support the customer's chosen mix of network protocols and WAN links at a high level of performance. Isolation of local traffic at the subsystem level must occur, while wide-area links must be able to support the necessary traffic without causing a "bottle-neck".

Flexibility: Today's solution must be extensible to support rapid network growth and increasing network complexity as new subnets are grafted onto the existing topology. Protocol-independent bridges and multi-protocol routers will need to work together in this new environment. In addition, a comprehensive framework for adding new network technologies, such as 100-Mbit/s FDDI, OSI routing or fast packet switching, must be integral to the current solution.

Reliability: To support mission-critical applications, the enterprise LAN must be highly reliable. The key factors used by customers to evaluate network reliability are:

1. Network components with built in reliability (high MTBF).

2. Powerful network management that quickly isolates the source of network problems.

3. Network control algorithms, such as the IEEE spanning-tree algorithm, that automatically re-route traffic around failed WAN links.

Low cost of purchase & ownership: If the above criteria are met, customers will choose the least costly alternative, both in terms of initial purchase price and ongoing network support.

Preservation of existing investment: Customers prefer solutions that allow them to continue using their existing equipment. In addition, the expected useful life of the new equipment to be purchased is a key consideration.

"Functional Modularity" is a key implementation concept necessary to grow a Global Enterprise LAN. Each functional component in the network, ie. hub, bridge, router, etc., should be a separate modular product. The individual modules can be flexibly combined to create optimized network topologies, (compared to the chassis approach offered by some LAN hardware vendors). The modular, single-function products provide many benefits analogous to the component approach popular in high fidelity sound and video electronics products.

Integrated Network Management: Integration of all key network components under a single network management system is key to a global enterprise LAN strategy. Powerful, integrated network management must deliver the total network visibility and control critical to effectively monitor and isolate faults in the network.

Standards Support: Support for industry standards is essential to multivendor interoperability and long-term investment protection. The vendor you choose must be firmly committed to help establish and support critical LAN networking standards from IEEE, ISO, and ANSI.

Technology Partnerships: Selected technology partnerships with other companies is expected to help broaden a vendor's product line and accelerate the delivery of new products.

Transport Infrastructure and Network Services: An total enterprise LAN solution is constructed by combining solutions to smaller segments of the overall networking problem. This brief focuses on the transport infrastucture -- the network connectivity through layer 3 and 4 of the OSI Reference Model -- required for the enterprise LAN. The role of this transport infrastructure is to transparently filter, forward, and route data for upper level network services. To further clarify, transport infrastructure solutions tend to focus largely on hardware products, such as hubs, bridges and routers, while network services solutions are software-oriented products, such as LAN Manager or Novell NetWare. The transport solution must support the customer's mix of network services solutions.

Enterprise LAN Solution Areas: There are four key solution areas in the overall enterprise LAN solution. These solution areas and transport infrastructure products are described below. Of course, these transport infrastructure products must be compatible with leading network services solutions, such as ARPA services, HP NS, DECNET, Novell NetWare and LAN Manager.

1. Solutions for Personal Workstation to LAN Workgroup Connectivity

This solution area includes the products required to connect personal workstations, such as PCs and UNIX workstations, and servers together to form workgroup LANs.

The transport infrastructure elements of this solution area include network adapter cards for the personal workstation, bus/star-oriented cabling schemes such as twisted-pair wire or thin coax, and wiring concentrators or hubs. Vendors must design to meet the customer requirements described above. The product lines must also offer a highly scalable, economical LAN workgroup solution. Specific LAN test instruments must be provided to simplify the previously cumbersome task of verifying the suitability of existing cable and LAN components.

In addition to the above transport infrastructure products, vendors must also offer a full range of network service solutions software for creating LAN workgroups such as LAN Manager, Novell NetWare, and ARPA Services for the PC, and network services for the PC.

2. Solutions for LAN Workgroup to LAN Workgroup Connectivity

This solution area includes the products required to interconnect LAN workgroups, located together on one site and/or at a series of remote sites, to form the enterprise LAN. A series of interconnected (or internetworked) LAN workgroup subnets is called an internet.

The transport infrastructure elements of this solution area include local and remote bridges and routers, and network servers with multiple LAN and/or WAN links. For fairly simple, single-vendor networks where high internet performance is not critical, internetworking solutions based on network servers, such as Novell NetWare servers with WAN links, are often sufficient. For higher-performance, multivendor networks with straightforward topologies, local and/or remote bridges are often the best solution since they offer true "plug-and-play" operation and an excellent price/performance ratio. For large, complex or mesh topologies, multivendor networks requiring a high degree of manageability, routers are the preferred solution because they offer greater reliability, security, and network control.

3. Solutions for LAN Workgroup to Mini/Mainframe Computer Connectivity

This solution area includes the products required to link LAN workgroups to minicomputers and/or mainframes. Defined in terms of the shifting computing environment, these products connect the new computing environment (internetworked LAN workgroups) to the old environment (minicomputers & mainframes) or incorporate existing computers into the new environment as servers. Two main sub-categories are of primary importance.

Mainframe/minicomputer terminal connectivity solutions: For quite some time many applications will continue to run in the mini/mainframe environment. Users in the new workgroup environment need access to these applications in addition to newer applications developed for the new computing paradigm. This is often accomplished using terminal emulation software on the personal workstation and one of three connectivity options: 1) direct connection of individual workstations in the LAN workgroup to the mini/mainframe using terminal adapter cards that are often built into the personal workstation, 2) shared connectivity over the LAN to the terminal controller via a LAN/terminal controller gateway, or 3) direct network access to the host via the LAN, usually referred to as virtual terminal. The third approach is the most desirable because it only requires the addition of software. The most widely available service to do this is ARPA's TELNET service. It is available on most host systems (even some IBM systems) and workstations. This approach works well for applications that are character-oriented. For block mode applications such as those on HP 3000 systems, HP offers a block mode virtual terminal for UNIX and DOS workstations.

Mainframe and minicomputer incorporation as a server: As we've already mentioned, much of today's data is resident on mini and mainframe systems. In most instances it is not desirable or maybe even possible to relocate the data and the application that maintains it to a new platform, specifically a workgroup LAN. It is far more desirable and reasonable to integrate the mini and the mainframe as servers to the new platform. This can be accomplished by integrating the LAN workgroup and mainframe/minicomputer environments by incorporating the workgroup networked operating system (NOS) on the mini/mainframe. From a transport infrastructure viewpoint, it is necessary to support the appropriate protocols for the common NOSs. Once having done this, the mini/mainframe can now fully participate as a server to multiple workgroup environments.

Usually both terminal/application access and the server integration approaches are used together to provide the necessary functionality. As an example, a workstation might use TELNET to access a host application while using the same host system as a server for high-speed printing using the print spooling feature of the workgroup NOS. These types of products allow data to be retrieved from mini data bases and placed into PC applications for analysis and presentation. Such transition products are necessary to help customers evolve to the new computing environment without sacrificing or replacing the effective mini/mainframe applications they already have in place.

4. Network Management

This last solution area includes the products required to manage the enterprise LAN. Specifically, transport infrastructure network management products include the management agents built into the components of the network and network management stations. Network management for the transport infrastructure should allow rapid identification and resolution of network problems to help provide highly reliable network transport.

Most networking vendors are following industry standards in this area and are standardized on SNMP network management agents for most of their LAN infrastructure products. These products will need to be based on both MS DOS and Unix platforms to offer customers a full range of price and performance options.

CONCLUSION

The best way to create a global enterprise LAN is with extended LANs as opposed to more conventional backbone technologies. However, these extended LANs will need complement, "tributary" networks using X.25 and ISDN. This will allow customers to build true global infrastructures based primarily on extended LANs but using lower cost and lower bandwidth options in those areas of the world where these technologies are more appropriate.

PC-based LANs are an increasingly important utility for corporations. And now the strategy for PC-LANs is being shaped by the influence of new products and the compatibility with existing equipment. Following the path to a global enterprise LAN will ensure efficient and transparent connectivity between your islands of information.

UNIX is a registered trademark of A.T.& T.

HOW CAN NETWORKING ON MPE XL SYSTEMS HELP YOU MEET YOUR BUSINESS NEEDS

GREGG KORNFELD
HEWLETT PACKARD
INFORMATION NETWORKS DIVISION
19420 HOMESTEAD ROAD
CUPERTINO, CA 95014

INTRODUCTION

Networks are being utilized by businesses as a key competitive weapon in their day-to-day operations. Corporations are discovering new ways to use communications and computing technologies to improve their profitability - both by reducing expenses and by increasing revenues. This paper shows how the networking offering on the HP 3000 enables these competitive advantages.

BACKGROUND

HP is a leading producer of networking products, services, and capabilities that deliver computing power for information management. HP's computing philosophy is based on the concept of distributed processing. Distributed processing is the general term used to describe the placement of computer, whether that is on the floor of a manufacturing facility, in an accounting department, in a laboratory, or in a corporate office.

Organizations gain a number of benefits from arranging their computer systems into networks. Networks enable all computer users to take advantage of the organization's total computing capabilities through information exchange and sharing of resources. Small systems can access the resources and greater computing power of large systems, while large systems can off-load applications best handled by personal workstations.

The networking environment provides several key capabilities, including transaction processing, which may involve communications over both short (Local Area Network) and long distances (Wide Area Network); PC integration that enables users to leverage departmental resources; and voice/data/text integration in both local and wide area networks.

HP's network solution, called HP AdvanceNet, provides the following benefits:

Multivendor Connectivity -- HP AdvanceNet is based on defacto and international industry standards. A networking solution based on standards enables effective communication between a variety of computer systems. This protects your current computer equipment investment and gives you the flexibility to choose the best technologies and applications for your future needs.

Scalability -- HP AdvanceNet is designed to meet your needs today and will grow with your company to solve tomorrow's problems. HP is committed to helping your organization easily accommodate growth through scalable solutions that simplify network expansion. The depth and breadth of HP's product line allows you to choose the most cost-effective solution for your current needs, while assuring compatible growth as your company grows.

Performance -- HP realizes that the system CPU must be devoted to critical on-line transaction processing. As HP 3000 systems evolve toward higher and higher MIPs and transaction rates, network performance will improve. In addition, the percentage of the system CPU devoted to networking will continue to decrease. This will ensure minimized user response time.

High Availability -- HP realizes that network downtime can mean lost business. HP is recognized as a leading supplier of high quality, reliable computer systems and networks. In addition, HP's network management products, along with our top-rated support, help ensure maximum uptime and optimal performance. HP also provides network problem isolation and management of problem isolation in a multivendor networking environment.

PC Integration -- HP continues to provide leading PC integration. This provides you with the best of both worlds: access to a wealth of PC applications plus access to the power and resources of HP minicomputers. HP also provides connectivity to other vendors' PCs and PC LANs and supports LAN Manager, which is an emerging standard for PC networking; this preserves your current PC/PC LAN investment.

The relationship between these networking benefits and your business goals depends on a variety of factors, including the type of industry, the company size, and your competitive position.

ZETA COMPANY: A Case Study

As an example of HP 3000's network capabilities, consider the hypothetical Zeta Company, which uses networking in its day-to-day operations to stay competitive.

Zeta Company operations has:

* A centrally located headquarters in New Jersey with multiple HP 3000s (MPE V & XL) and NON-HP systems.

* Many remote sales and distribution sites across the United States.

* A parent company in San Francisco running IBM mainframes.

Zeta Company is under increased competitive pressure to reduce operating cost and increase the flow of information.

1. More and more users need access to critical On-Line Transaction Processing (OLTP) applications.

* Many of Zeta's remote sites have existing PC LANs that require system integration.

* Users in the central site require access to both the HP and NON-HP systems.

* The parent company is investigating ways to integrate into Zeta's E-Mail network.

* Users at the remote sites require access to the central site. Response time for the users on the central site should not be compromised.

2. Zeta Company must reduce operator costs at the remote sites, without compromising system availability.

3. Zeta Company is under increased pressure from the parent company to reduce overall computing costs; this means controlling networking costs.

= = > HP 3000 networking will solve these critical issues facing Zeta Company; lets see how!

The NEW DTC

The New DTC provides both terminal and X.25 connectivity to the HP 3000 system. Initially, the DTC stood for Distributed Terminal Controller, and provided single system access to an MPE XL system. With enhancements to the DTC, it now provides multivendor access to multiple systems, as well as X.25 connectivity to the MPE XL system. Due to the new capabilities added to the DTC, the name has been changed to the Datacommunications and Terminal Controller.

Local Terminal Connectivity

Local users attached to a DTC can now switch between multiple systems on the LAN. There is no need for using virtual terminal to connect to multiple machines. Now, a user can get, high performance access to applications on multiple machines and can even easily switch to a backup system in the event of a failure of their primary machine.

Today, the DTC provides direct access to MPE XL systems and it can also be turned around (using its RS-232 ports) to switch between other systems such as MPE-V and DEC. This kind of flexibility really improves end-user productivity.

In the first half of 1991, ARPA Telnet will become available on the DTC. This functionality will allows users directly connected to the DTC to access applications on systems that support the ARPA Telnet/Ethernet stack. In addition, users connected to NON-HP systems will be able to access MPE XL applications by using the DTC as a protocol converter. By choosing this implementation instead of a system-based implementation, HP has created a Telnet front-end processor for MPE XL. This provides the most efficient and productive way for remote users to access HP 3000 applications from a Telnet host.

How does the new DTC help Zeta? Lets look closer at Zeta's central site. The central site is made up of three HP 3000 systems and one NON-HP system. Zeta has already successfully migrated four of their MPE V system to two Series 960s. Zeta still requires a Series 70 for final porting of applications to the XL system; once the port is complete, the series 70 will be migrated to a Series 949. Users require access to all systems.

Today, Zeta can put in place a DTC infrastructure that will allow all of its users to transparently access all of its systems. As the last MPE V system is migrated, the DTC ports used for MPE V access can now be used for additional MPE XL users. This provides investment protection for Zeta during its migration. Lastly, when Telnet capabilities are added to the DTC (first half 1991), direct system access will be provided for all systems. Zeta's investment in DTC hardware is always protected.

Remote User and System Access

HP has taken a different approach with MPE XL, and implemented the X.25 link in the DTC. This has a number of important benefits including the ability to share an X.25 link between multiple systems, and dramatically improve performance. Relative to NS X.25 on a Series 70, X.25 on a 950 has shown up to 75% MORE throughput, while consuming up to 85% LESS CPU on the XL host. And for X.25 PAD terminal connection, the DTC solution is six times more efficient than host-based PAD on MPE V systems. This translates to more MPE XL CPU left for running your business, and less is spent on networking.

For the Zeta Company, remote sites are connected to the central site over an X.25 network. Many of the remote sites have system-to-system communication needs. Some of the sites only require remote end-user access. By having the HP 3000 systems share the same X.25 card, Zeta is able to reduce networking costs at the central site; a major goal for Zeta management. Zeta is also able to effectively integrate remote end users, without having to incur additional CPU overhead on the XL hosts at the central site. This allows Zeta to meet another corporate objective of increasing information flow without decreasing user response time.

NEW IBM CONNECTIVITY

Remote Access by IBM Users

IBM mainframe users can now take advantage of the power of MPE XL applications. With HP Distributed Host Command Facility, or DHCF, mainframe-attached users on IBM 3270 terminals (or PCs emulating 3270s) can use their existing equipment to access VPLUS applications, product from HP's Value Added Business partners, and HPDeskmanager. This access is provided without any special software on the mainframe. This approach is easier for HP, IBM, and you to support.

What about performance? On a single 64kbps link between an HP Series 955 and the mainframe, HP has benchmarked better than three second response times on over 30 concurrent sessions. Overall, HP has seen in excess of 30 transactions/minute in this configuration, and adding additional links between the HP 3000 (MPE-XL) and the mainframe increases this number even further. DHCF allows a broader range of people in your company to benefit from access to the information and applications on the HP 3000 while protecting 100% of the investment in IBM terminals and wiring. It also allows you to realistically consider using some of the less expensive HP 3000 MIPs to offload applications that might be considered for use on the IBM mainframe.

For Zeta, users in the parent company will now be able to integrate into Zeta's HPDeskmanager network without having to invest in any new hardware.

New IBM Products Allow Standardization of Backbone Network

With Release 2.1, new IBM connectivity products allow you to make the choice of supporting and managing a single backbone network for both HP-to-HP, IBM-to-IBM, and HP-to-IBM communications.

If you want to standardize on an X.25 network, the SNA/X.25 Link/XL (SNA over an X.25 backbone) product enables HP 3000 Series 900 systems to exchange data with IBM systems over an X.25 network. If you want to standardize on a SNA network, the NS over SNA/XL (NS over a SNA backbone) product provides the capability to allow the NS3000/XL Network Services to run over an IBM SNA network for HP 3000-to-HP 3000 communications.

X.25 and SNA have emerged as the most important wide-area networking alternatives. Though X.25 is a standard, SNA continues to be a prevalent networking architecture. In some cases, the need for multiple, separate backbone networks are necessary to support both HP and IBM communications. However, as networking costs rise, it is increasingly difficult to maintain and manage two separate networks. SNA over X.25 and NS over SNA are positioned for those environments that want to standardize on one backbone network.

One backbone network provides many benefits:

Reduced Costs -- Phone line costs and networking costs are reduced by requiring only one card and access line to the network.

Easier Management and Control -- One backbone is easier to manage than two, and the number of operators and equipment to manage is reduced.

Investment Protection -- Existing backbone network is leveraged.

If your environment is primarily IBM, then NS over SNA provides an economical way to utilize your existing network. If your environment is multivendor and includes a variety of systems, then SNA/X.25 Link/XL allows you to standardize on the OSI/ARPA standard -- X.25.

With enhancements to the SNA/SDLC Link/XL, the HP 3000 system can now provide link-level alerts to IBM's network management product, NetView. These delayed alerts provide even stronger integration into an IBM SNA backbone.

Zeta's environment is primarily X.25 with only one SNA connection from the central site to the parent company. By using the SNA/X.25 Link/XL product, Zeta will be able to standardize on an X.25 network. Zeta will save costs by no longer requiring two network access lines. Zeta will leverage the existing X.25 line for both NS and SNA traffic. And, with the multiple system capabilities of the DTC, the parent site has direct access to all HP 3000 systems at Zeta's central site, this will yield a big reduction in CPU overhead for the HP 3000 system that was acting as an SNA gateway. Zeta will also achieve increased control and efficiencies by now only managing one backbone network.

PC-to-HP 3000 Integration

HP is a recognized leader in integrating PCs with HP minicomputers. In fact a recent Gartner Group survey rated HP number one in client-server PC minicomputer integration. Products like HP OfficeShare and Business System Plus helped to establish this strong reputation which HP is continuing to fortify with new products and directions.

There are three main areas in PC-to-HP 3000 integration that HP is extending. They are as follows:

1. Connecting to industry leading PC LANs.

2. Evolving the OfficeShare functionality to the industry standard, HP LAN Manager.

3. Creating value-added applications from HP and through third parties that build on the strengths of networked PCs and the HP 3000.

With the large installed base of Novell LANs and with the wide industry acceptance of LAN Manager, these PC Network Operating Systems (NOS) will have broad usage.

Novell LAN Connectivity

Many HP customers have a large investment in PC networking. These customers use Novell's NetWare products running on PC class servers. The NS LAN Gateway provides a full suite of data and resource sharing capabilities between a Novell client PC and the HP 3000. The HP NS LAN Gateway brings the power of MPE applications and NewWave Office to PC users on Novell LANs. Novell client connectivity has also been enhanced with the recent introduction of Portable NetWare on the HP 3000 (MPE-XL) (available first quarter 1991). Portable NetWare is based on Novell's latest and most powerful product to date, NetWare 386. NetWare for the HP 3000 (MPE-XL) will also support all of these features. Portable NetWare provides file and print sharing, server administration and management, and Sequenced Packet Exchange (SPX) protocol for developing client-server applications.

LAN Manager Connectivity

HP 3000 support of HP LAN Manager clients offer a migration path to existing Office Share customers. HP LAN Manager supports the industry standard TCP/IP which is used in many enterprise-wide networks. HP has made the commitment to provide LAN Manager protocols on all of its computer platforms. This benefit will allow you to choose the platform appropriate for your application, and feel secure that you will be able to migrate to more powerful computers as needs dictate.

In the first half of 1991, the HP 3000 (MPE-XL) will support Application Program Interface (API) Named Pipes. Named Pipes is a method used by application programs to talk to each other and to exchange data. This exchange of data is, of course, the foundation on which client-server computing is based. Because Named Pipes is the API of choice for all LAN Manager client-server implementations, support for it on the HP 3000 (MPE-XL) is very important. The benefit to you is interoperability and portability. With a common API, LAN Manager clients from one vendor will be able to talk to LAN Manager servers from other vendors.

Zeta has remote sites which have Novell PC LANs that have not yet integrated the HP 3000 system. At one particular site, the Novell NetWare LAN has grown rapidly within a department, and now needs access to the information in the ORDERS database on the HP 3000. It may not always be cost effective to replace the NetWare LAN with OfficeShare or LAN Manager, but with the NS LAN Gateway, users can continue to access the NetWare server as they had before, plus they can use the power of NewWave Office to extract information from the HP 3000 database for use with their PC applications. The NS LAN Gateway allows access to any HP 3000 application. These users can now get linked into the HPDeskmanager network and communicate more effectively with the rest of the company. The bottom line is that you have made a department of people in your company more effective by giving them access to information and service on the HP 3000 without having to sacrifice an existing investment in LAN hardware and software.

Zeta also has a remote site that is currently using PCs in terminal emulation mode only. In an effort to evolve to a client/server environment, Zeta is considering a PC LAN. Using HP LAN Manager clients, Zeta will be able to have access to NewWave Office and continue to have access to all HP 3000 applications. Zeta's investment in HP LAN Manager clients is also protected as the industry develops more software applications based on the API Named Pipes.

Network & System Management

HP OpenView System Manager enables system operators to monitor and control multiple networked HP 3000s from a single, centralized console. It is targeted for operating environments that wish to lower operator costs while maintaining a high level of service. This is accomplished by centralizing operation control and expertise. For remote sites, this provides the foundation for unattended operation support. The consolidation of system control is the first step toward fully integrated system management functionality under HP OpenView.

HP OpenView System Manager provides control of applications and systems in all areas of data center management across the network. It provides exception-based system management, where operators are notified of system events graphically and automatically via the HP OpenView network map. This frees them from the tedious task of monitoring console messages for problems. It allows the operators to define which events will get reported. It provides flexibility and allows the operators to eliminate unwanted notifications. The virtual console feature provides full system control, including the ability to shut down and restart systems. The event application programming interface allows third-party and user-written applications to report events to the System Manager console.

OpenView System Manager provides you many benefits including:

<u>Reduced Operation Costs</u> -- HP OpenView System Manager reduces operator cost by minimizing the operations expertise required at remote sites.

<u>Increased Efficiency</u> -- HP OpenView System Manager increases operator efficiency by allowing operators to send commands to and receive responses from multiple system simultaneously. It frees operators from monitoring console messages by notifying them when problems have occurred on their systems.

<u>Increased System Availability</u> -- HP OpenView System Manager provides faster fault detection by immediately displaying managed system events. HP OpenView System Manager also increases system uptime by providing faster error resolution, and improves the level of service by making centralized system management expertise available to all systems.

Reduction of operation costs at remote sites has been a priority for Zeta's management; however, concerns have been raised over compromising system availability. By using HP OpenView System Manager, Zeta will be able to reduce operation costs by centralizing operations and control at the central site. Zeta will not compromise system availability, but instead, it will increase it. HP OpenView System Manager will allow Zeta's staff faster fault detection and faster error resolution, which means increased system uptime. With the exception-based management of systems, Zeta's central staff will become more efficient, and will now be able to take on additional responsibilities. HP OpenView System Manager helps Zeta meet their business objectives.

SUMMARY

HP 3000 networking products provide you many benefits that will allow you to stay ahead of the competition. Zeta, like many organizations, was under pressure to increase the flow of information and to keep operation costs down. With the DTC, Zeta was able to increase connectivity of both local and remote users to HP and NON-HP systems. The new IBM products provided Zeta with tighter integration into their parent company's IBM backbone network. And, by Zeta standardizing on an X.25 network, they were also able to decrease networking costs by eliminating a network access line. The PC-to-HP 3000 integration products have provided Zeta with an evolution path to a client-server environment. Lastly, Zeta was able to reduce operations staff at key remote sites by implementing HP OpenView System Manager. By providing a centralized console, Zeta was able to meet cost objectives for the year, and at the same time, improve efficiencies at the central site.

How to be a "SMART" User of a "DUMB" Terminal

By

Michael Hornsby

Beechglen Development Inc.
2026 Beechglen
Cincinnati, Ohio 45233
(513) 922-0509

For

Interex HP Users Conference
Paper #7064
Boston
August 20-23, 1990

Abstract

Many of us spend 1,000 to 2,000 hours in a year working at a terminal or a PC emulating a terminal connected to an HP3000. These so called "Dumb" terminals have many features that can make interacting with the HP3000 a more pleasant and efficient experience.

Some of the most basic and frequently asked questions regarding the HP3000 revolve around terminal usage such as: When do I use the Carriage Return versus the Enter key? Why do I have hit carriage return before typing in HELLO? Which is better: character or block mode? Understanding the answers to these questions is basic in beginning to effectively use the HP3000.

The purpose of this paper is to examine features and modes of terminal operations with examples of practical everyday uses.

I. Introduction

With the introduction of the 2647 display terminal, Hewlett-Packard produced one of the best documents for understanding how the terminal communicates and interacts with the HP3000. Today's terminals are accompanied by a very slim users manual that contains minimal useful information. The typical user learns by trial and error how to use the terminal in interacting with the HP3000. This situation leads to frustrated users, programmers and system managers interacting with the HP3000 in an efficient and unproductive manor.

This paper documents how a terminal user interacts with the HP3000 and provides tips and techniques for being a more proficient user.

A. Modes of terminal operation

Every terminal functions as an input and output device. The keyboard is the input device and the display is the output device. Some characteristics of how the terminal input device and output device operate are changed depending on the application desired. The terminal can operate in one of three modes: local, character and block mode.

1. Character mode interactions

Character mode is the default mode of operation for most terminals. In this mode each character typed at the keyboard is sent to the HP3000 and the ATP or DTC port on the HP3000 echoes back the character to the terminal display.

HP3000 CHARACTER MODE OPERATION

```
+----------+                              +----------+
|          |    EACH CHARACTER TYPED      |  HP3000  |
| Keyboard |>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>|   PORT   |
|          |   IS SENT TO THE HP3000      |  INPUT   |
+----------+                              ////////////

+----------+                              \\\\\\\\\\\\
|          |      AND ECHOED BACK         |  HP3000  |
| Display  |<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<|   PORT   |
|          |   TO THE TERMINAL DISPLAY    |  OUTPUT  |
+----------+                              +----------+
```

Each session on the HP3000 starts in character mode. The terminal user enters a carriage return and the HP3000 responds with a prompt (usually a :) and a terminal read expecting the HELLO command.

One of the most basic first questions from new users is: Why do I need to enter a carriage return to get a : before I can type HELLO?

It is very frustrating to type in HELLO USER.ACCOUNT and then have the HP3000 respond with a : prompt. What actually happened is that everything up to the carriage return was ignored!

The answer is the HP3000 uses the carriage return to determine the speed of the terminal and to start a session. Each port can be configured as speed specific or speed sensing. Speed specified ports are desirable for installations where the terminal will be connected via modems or multiplexors. Speed sensing is desirable for directly connected terminals. In either connection method it is best if the user can communicate with the HP3000 at the highest possible speed.

It is an amazing fact that most direct connect terminals are configured to run at 960 characters per second. The actual maximum speed is 1920 characters per second (excluding obsolete ATC and ADCC based HP3000's).

After that initial carriage return every interaction with the HP3000 is controlled by a program running that issues reads to the terminal and reacts to the content of the buffer returned from the completed terminal read. Terminal reads can be completed based on one of three criteria: The user entered a carriage return (or some other redefinable character), a certain number of characters have been entered, or a read timer set prior to issuing the terminal read has expired.

```
                 |        User         |      System       |
                 |        Think        |     Response      |
                 |        Time         |       Time        |
                 |                     |                   |
     Time >----|--------------------|------------------|---->
                 ^                     ^                   ^
                Read                 Read                Read
               Posted              Completed            Posted
```

Once the terminal read has been completed the elapsed time until the next terminal read is posted is called response time. Actual response time has five components: transmit time, cpu time, disc IO time, wait time and receive time.

Transmit time is the amount of time it takes to send the data to the HP3000. This is minimal for character mode but for other modes it can be more significant.

CPU time is the total amount of processing time required to perform the programmed task between when the read completes and the next read is issued. Sometimes this is a very small amount of time, for example storing the data entered in a buffer. Sometimes this is a very large amount of time, for example calculate Pi to 100 decimal places.

Disc IO time is the total amount of time waiting for Disc IO operations to complete. Sometimes this is zero if the program didn't need to read or write any records. Sometimes this time spent is greater than the amount of CPU time ie for a report request.

Wait time is the time that the program waits for system resources such as memory, file locking etc.

Transmit time is the amount of time it takes to display the results of the terminal read. This time is often a very significant part of the total response time and is directly effected by the operating speed of the terminal. Anyone who has dialed into an HP3000 at 120 characters per second can attest to the unproductive time waiting for the screen to paint.

In character mode applications the programmer is totally responsible for painting the screen and cursor positioning. This tedious coding is usually done in a generalized section of code called a screen handler. The main drawbacks to character mode operation are that each field on a screen is treated as a separate terminal read, and response time between fields can be erratic.

On the MPE/XL systems every terminal is actually connected via a local area network (LAN). Applications that perform short character IO (reading and writing 1 to 10 characters) has proven to be a major migration and performance issue.

Local area networks packetize the information sent and received. Sending very small packets can cause excessive overhead especially when multiplied by a hundred or more users.

Another common question regarding character mode is: Why do I have to use the BACKSPACE key instead of the arrow keys?

The answer is that in character mode when we type, the characters are echoed and buffered. A BACKSPACE tells the HP3000 to backup one character in the buffer. The arrow keys do not send any data to the HP3000 and therefore when we use the arrow keys the buffer is not backspaced even though the cursor is moved on the terminal screen.

These arrow keys are very useful and can provide a large improvement in user productivity. At the end of this section we will develop the topic of how to use the arrow keys in character mode.

When moving in between fields character mode screen applications require the application program to read the data and reposition the cursor at the proper position for the next field. This requires that the program be scheduled and dispatched for each field entered. The response time from field to field can therefore vary widely.

A very interesting side effect of this is that systems whose applications run mostly in character mode have a greater memory requirement. This is because their code and data areas (segments for MPE/V and spaces for MPE/XL) are referenced for each field on each screen. Systems with insufficient memory tend to thrash resulting in very erratic response times.

2. Block mode interactions

Enter V/3000, on each HP3000 Hewlett-Packard provides a data entry and forms management package that has implemented block mode terminal data transfers. V/3000 provides a method for performing data entry into a 'flat' file and a programming interface for getting data into a data base or KSAM file.

In block mode the terminal screen is formatted so that the user is restricted to typing only in certain areas called unprotected fields. The screen is formatted into protected and unprotected fields. Protected fields are for text, headings and display only fields. Unprotected fields are the only place that data can be typed in from the keyboard.

In block mode the screen is painted once and data is typed into the unprotected fields. Only after all of the appropriate fields are entered does the user use the enter key to send data to the HP3000. In block mode each character is not sent and echoed to and from the HP3000 port.

HP3000 BLOCK MODE OPERATION

```
  +----------+
  |          |     EACH CHARACTER TYPED IS
  | Keyboard |     LOCALLY ECHOED TO THE
  |          |     TERMINAL DISPLAY
  +----------+
       v
       v
       v
  +----------+                              \\\\\\\\\\\\
  |          | AND SENT FROM THE DISPLAY |  HP3000    |
  | Display  |>>>>>>>>>>>>>>>>>>>>>>>>>>>|  INPUT     |
  |          | MEMORY AT ENTER OR FKEY   |  PORT      |
  +----------+                           +----------+
```

When an entire unprotected field is filled the terminal automatically moves the cursor to the beginning of the next unprotected field. To manually move between fields the user must use the TAB or arrows keys. There is no interaction with the HP3000 until the ENTER key or one of the function keys is pressed.

One of the basic HP3000 questions is: When do I use the return key and when do I use the ENTER key?

The put off answer given most is: You use the return key for character mode and the ENTER key for block mode. This answer usually leaves the neophyte user in a very frustrated state wishing that the computer would know when to use the return key and when to use the ENTER key.

A better answer is to show the user a screen painted in character mode and a screen painted in block mode. Character mode screens require the computer to position the cursor after the user uses the return key. On block mode screens the unprotected fields are usually in inverse video. When the terminal is in block mode hitting the return key causes the cursor to be positioned in column 1 of the current line. Hitting an alphanumeric key will cause the cursor to be moved to the next unprotected field.

It is not at all unusual for a given user to have to operate in both character and block mode applications on a daily basis. When designing and integrating applications the most important thing is to keep the user interactions as standardized as possible.

Since there is no interaction with the HP3000 in between fields the user can develop a sense of timing for entering data on the screen. When the ENTER key is pressed the application program reads the contents of the unprotected fields.

The main drawback to block mode screens is that field editing is done after the user has entered data in all of the fields. In character mode each field is usually edited as it is entered.

Another drawback to block mode applications is the operating system's dispatching and memory management algorithms. In character mode, the dispatcher considers each terminal read for each field a new transaction, and thus starts it out at the base of the CS queue. In block mode the ENTER is counted as one terminal read and the process's priority starts at the base of the CS queue and then degrades to the limit of the CS queue where it must compete with other CPU intensive operations such as interactive compiles and report requests.

Remember that in character mode the processes code and data areas are being kept in memory because each terminal read must reference these areas. In block mode these areas will be swapped out if memory pressure is being caused by IO intensive jobs or sessions. This is why block mode response times can be impacted by running batch jobs during the day.

The result is that each time the ENTER key is hit the memory manager must fetch in the code and data areas for this process. The only solution is to modify the V/3000 programs to perform timed terminal reads. This is a relatively new feature of V/3000, so not many applications make use of it.

Timed V/3000 reads have the added benefit of allowing the application program to check and see if it should shutdown or display a message from the operator. On a large system it is quite impractical to call each user up and ask then to sign off.

Most sites simply use the ABORTJOB command to get everyone logged off prior to a system backup or shutdown. This leaves the terminal in block mode in an inconsistent state. The screen is still formatted with protected and unprotected fields but the application is no longer running and the terminal is signed off. This forces the user to perform a terminal reset to clear the screen prior to starting the next session.

Local mode is the least used mode and also the least understood. Most users arrive in local mode by accident by getting to the MODES function keys and hitting the REMOTE function, so that the * is turned off. This tells the terminal not to send any data typed to the HP3000 but instead to display the characters typed on the screen. This is a very useful feature at times and the next section develops these uses in detail.

B. What are escape sequences?

When dealing with terminals or printers it is inevitable that the term escape sequence will eventually come to the surface. Escape sequences are commands to the terminal or printer instructing it to perform a specific function. They are a programming language all to themselves and can vary in operation from terminal to terminal.

Escape sequences derive their name from the fact that they start with the ASCII ESC character and must follow a vary precise set of syntax rules. These sequences are used for tasks such as positioning the cursor, clearing the screen, etc.

One of the distinct advantages of using V/3000 or a screen driver is that the application code that performs terminal IO and therefore must issue a multitude of escape sequences, are localized and therefore can be maintained separately. This is important because as new terminal features are introduced new escape sequences are required to drive them.

In the later part of this paper we will go into some of the more useful escape sequences and provide hints and techniques for programming and debugging them.

C. General Terminal Features History

In examining the HP3000 world one of the most confusing and perplexing end user, applications development and support issues has been the evolution of terminals. Any system with more than a dozen users has more than one type of terminal in use. Although these terminals are basically compatible, their individual differences have caused a great deal of consternation, confusion and confrontation.

In the past ten years Hewlett-Packard has gone through five different terminal families! Each family has contributed in either a functionality or a price breakthrough. But end users must adapt and learn the quirks of their particular terminal. Applications programmers must change and maintain programs when new terminals are introduced and system managers must learn how to support and connect new terminals to the HP3000.

Those of us who have been around the Hewlett-Packard systems for some time fondly recall the genealogy and evolution of Hewlett-Packard computer terminals. At a recent meeting of HP3000 types one discussion wandered into reminiscing the past, and person after person pulled their key ring out of their pockets and produced the key that was required to open the older models of Hewlett-Packard terminals. One person even had went so far as to have his key gold plated!

When I first joined Hewlett-Packard every SE was sent to a data terminals class at the now defunct data terminals division. This was the only Hewlett-Packard class that had a pretest and if you failed, they sent you home. The interesting part was that most of the material in the class was about the 264X and 2700 terminals that at that time were obsolete by new terminals. The courseware never did catch up with the current technology.

Hewlett-Packard Terminal Development History

```
Time >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Present
---------------------------------------------------------------->
264X      262X      2700      1XX      239X      VECTRA      700/XX
```

The X denotes a family of terminals with a range of sub models, generally the lower the number, the less functionality and the higher the number the greater number of features. Also, in Hewlett-Packard tradition most terminals had a letter appended to the end of the family and model number. This practice goes back to Bill and Dave's first product the 201B. They didn't want their customers to think that they only had one product.

264X - This terminal family was Hewlett-Packard's mainstay for many years. Many would only recognize the 2647F which Hewlett-Packard mandated as the console for the HP3000 series 68 and 70. These terminals were the first to feature block mode and had an optional printer interface. Later models also supported a BASIC interpreter, graphics and word processing applications.

264X hints:

#1. Check your hardware support contracts for any 264X models. If they are on support you should hunt them down and verify that they're still in use.

#2. These terminals were notorious for collecting dust and dirt inside. If you have been using one of these terminals for a number of years you should open it up and see if it needs to be cleaned out. By the way, blowing the dirt out is preferred to vacuuming.

262X - This line of terminals was housed in an ET type of configuration. (The CRT was on a pedestal and the whole thing distinctly resembled ET) The initial 2621 came in the A and P versions, the A version was standard and the P version had an internal thermal printer. Both of these terminals was character mode only.

The 2622 terminal supported both character and block mode and had a keyboard with a separate numeric keypad. Since the release of the 2622 all models ending with the number 2 have been the base models that support both character and block modes.

The 2624 terminal supported local forms caching and field edits. These features improved performance of block mode applications. Since the release of this terminal all models that end with the number 4 have supported these extra features.

The 2626 terminal introduced the concept of windowing. The terminal could be connected to two computers simultaneously and display up to four windows. It also supported a tone generator. The demonstration that came with the terminal emulated a juke box and could play a wide variety of songs. Unfortunately, no practical applications were developed that used the windowing feature. The W model was the initial terminal for HPWORD processing.

The 2623 and 2627 terminals supported graphic displays for DSG/3000 and HPDRAW. The 2623 was black and white and the 2627 was a color graphics terminal. Since then, you guessed it, the 3 indicates black and white graphics and the 7 indicates color graphics. Both of these terminals also supported the 2622 features.

The 2628 terminal was the climax of the 262x family. It supported black and white graphics, HPWORD, and all of the 2624 features.

262X Hints:

#1. Never pick one up by the top cover! Only two small screws hold the cover on and the bases of their mounts snap off very easily.

#2. The internal parts are very easy to interchange between models. Therefore if one breaks, try parts from a terminal that is working before pitching the terminal.

#3. The terminal keyboard cords have a tendency to short out. Before pitching the terminal try a keyboard from a terminal that is working.

#4. Many of these terminals had batteries that should be replaced and disposed of properly. A missing or dead battery will cause the terminal to use the default settings when powered on. This means that the terminal will be operating at 240 characters per second!

#5. The keys on the 262X terminals can get jammed through misuse. If you pull off the key that doesn't work you'll see two prongs sticking up. When the key is depressed these prongs should make contact. If they don't you can bend them in a little, and this should fix the problem.

This problem can manifest itself in different ways. One Sunday I was over a friends house and he gave me the nickel tour. He had a nice study set up in his spare bedroom which included a 2624B terminal, a printer and a modem. Once he had me trapped his ulterior motives became clear. He said that his terminal wouldn't work in block mode but character mode worked fine. After checking the configuration the problem became clear, the terminal's enter key didn't work. After further probing, it turned out that block mode stopped working after he started letting his four year old 'play' on the terminal. After carefully prying off the ENTER key I could see that the two prongs didn't touch when the key was depressed. I bent them together with a paper clip (after turning off the terminal!) and replaced the key. Presto. The terminal that wouldn't work in block mode now functioned perfectly.

The 2700 was a very expensive color graphics terminal that supported stand alone HPDRAW like graphics applications. This terminal was ahead of its time and not many were sold.

The 1XX personal computers, the 120, 125, 150, 110 all supported terminal emulation. The 150 and 110 supported both block and character modes. The 150 also supported black and green graphics and HPWORD processing.

The 239X family introduced a smaller footprint (took less desk space) and a decreased price. These terminals were similar in functionality to their older 262X counterparts.

The Vectra IBM PC clones introduced ADVANCELINK Hewlett-Packard's terminal emulator. ADVANCELINK emulates a 2624 terminal for the Vectra. Most IBM compatibles using the HP3000 are using REFLECTIONS from Walker Richer & Quinn, Inc. REFLECTIONS has two versions one that emulates the 2624, and one that emulates the 2627.

The 700/9X terminals are a low cost version of the 2622 and the 2624 terminals. These terminals were introduced to compete on price with the CUMULUS terminals. However the CUMULUS terminals also have a built-in telephone directory and calculator.

700/9X Hints:

#1. The initial models of 700/9X terminals have a class problem with power supplies that will burn out a capacitor and produce a lot of smoke. This always produces a lot of talk in the data entry department when an operator enters data so fast that he or she sets the terminal on fire!

If one goes you should specifically request that all of the terminals be repaired. This may save a trip from the local fire department.

The point of all of this discussion is that the Hewlett-Packard terminal line is very deep and many different terminals and personal computers are in use. The more varied your mix of terminals, the more time you will spend supporting them.

Support contracts should be reviewed periodically to delete any terminals that are no longer in use, and in many cases have been discarded. For the older terminals this can represent a large part of the overall HP3000 support costs.

Finally don't be afraid to attempt repairs on older terminals. Usually the problem is relatively simple and the terminal can be salvaged for continued use.

II. Local mode

Up to this point we have covered character and block mode operations in detail. The third mode of operation is one of my favorites and understanding local mode will allow the terminal user to really learn how to use the terminal effectively.

A. How terminal IO really works

When the terminal is in local mode any characters typed are simply echoed to the screen and not sent to the HP3000. This feature allows us to change anything in a terminal's display memory and then print it to a local printer or send it back to the HP3000.

When we are in local mode the local edit keys such as the arrows and the insert/delete keys are very useful for text editing. The secret is knowing how and when to switch modes. After a little practice it becomes very natural, and because of your terminal skills you will be a more productive HP3000 user.

B. The local edit keys

On all families and models of Hewlett-Packard terminals there are local editing keys that can be very handy for text editing. On the older terminals with the compact keyboards these keys are implemented in function keys. It is therefore recommended that before attempting any of the following local editing tricks that you use a terminal that has an expanded keyboard.

C. Uses for local mode operations

One of the simpler tasks that can be done with local editing is a redo of a previous command. Suppose that I'm using a utility or program that does not have a built in redo command, (EDITOR or SPOOK for example) and I wish to repeat the same command.

For example I'm in SPOOK and I wish to find an occurrence of a string:

```
>F@"Subtotal"
```

SPOOK will scan for the first occurrence of the string and display its line number and text. In this case I want to find all of the subtotal lines, so that I can add them up and check to see if my report balances without waiting for the printout.

Now suppose that I want to find the next "Subtotal". If I were a dumb user of a dumb terminal I would simply retype the same command over, and over, and over again to find each occurrence in the spool file.

Being a smart user of a dumb terminal, I would simply move the cursor to the > and hit the delete character key. This would remove the > from the line. Then I would hit the enter key. In character mode, hitting the enter key causes the terminal to transmit the entire line at once. This explains why we needed to remove the > prompt from SPOOK.

This also explains the strange phenomena that happens when you type in a command and then hit the enter key instead of the return key. As you type the command the terminal sends each character to the HP3000 and the HP3000 echoes it back to the terminal display. Then when you hit enter the terminal sends the current line to the HP3000 followed by a carriage return. The HP3000 actually has received the entire command twice!

What if we want to insert characters in the command instead of simply deleting them? Suppose we wanted to perform a find on "Account Subtotal"?

There are three methods for editing the command that was on the screen. The first method is to:

1. Put the terminal into local mode
2. Use the insert character key to turn on insert mode.
3. Position the cursor where we want insert characters
4. Type in the characters to be inserted
5. Use the insert character key to turn off insert character
6. Put the terminal into remote mode
7. Hit the enter key

This sounds like a very complex maneuver but with practice it comes very easily. An alternate method is to leave the terminal in remote mode but to trick the HP3000 into forgetting the characters typed when inserting. Thus the second method goes like this:

1. Use the insert character key to turn on insert mode.
2. Position the cursor where we want insert characters
3. Type in the characters to be inserted
4. Use the insert character key to turn off insert character
5. Position cursor on the line following the edited line
6. Hit the CTL and the X key simultaneously
7. Position the cursor on the line we edited
8. Hit the enter key

Again this sounds very complex but it works. Another alternate method is simply to use the backspace key to back up the HP3000 buffer instead of using CTL X to erase it. The only catch is that we have to remember exactly how many characters we typed in so that we can hit the backspace key the appropriate number of times. This also explains why the Hewlett-Packard terminals do not have a destructive backspace.

I use all three of these methods depending on the complexity of the editing required. The REDO command has been enhanced and added to many parts of MPE/XL but there are many utilities and subprograms that have do not have redo commands

One other situation where this type of editing comes in very handy is when using a line editor. Suppose we delete a line of text unintentionally. If the line is still on the screen we can get it back into the file very easily. Simply go into add mode, position the cursor on the line to be added and hit enter! Note, if the line had a line number displayed we would have used the delete key to clean up the line first.

This also can come in handy for saving file equations. Usually it takes some testing to get file equations set up for a new production application. Then we go into an editor and add these file equations to a UDC file for production use. Rather than retype the file equations we can perform a LISTEQ command which will list them out to our screen. Then we can go into editor and issue the add command. Using the same techniques as above we can capture each file equation.

There is one other area where local mode is very useful. Suppose that I want to do full screen editing on a part of a program or some other text. One method is to purchase a full screen editor, another is to list the text to be edited to the screen, and go into local mode to edit the text.

The tricky part is that we need a program that will read the screen contents back into a disc file so that we can join it into the original text. The utility PSCREEN which is usually used for printing the screen contents can also be used for this purpose. All we need to do is issue a simple file equation:

```
FILE PLIST=TEXTFILE,NEW;SAVE;REC=-80,,F,ASCII
```

Thus once we have the text the way we want it on the screen we can run PSCREEN which will copy it from the terminal display memory to a disc file named TEXTFILE.

This technique is also useful for just typing in a quick memo or note. I use it frequently for designing simple forms where being able to position the test on the screen makes the task much simpler.

II. Function keys

User definable function keys are available on most models of Hewlett-Packard terminals. They are useful for storing frequently used commands or responses so that they can be executed with a single key stroke.

Function keys assist the novice user to find the the right command for the job by trial and error rather than running to the manual for each command. They also help the accomplished user by minimizing the number of key strokes required.

I have a set of user defined commands that are driven by function keys that I used for years and have distributed widely. A listing of the UDC file is included in Appendix A and on the contributed library tape for this conference. The next section describes how these escape sequences work. They actually look more complicated than they really are. Once you set up one the rest come easily.

A. Loading pre-defined functions

Function keys are loaded with a set of escape sequences. The format for the escape sequence is:

```
ESC&f<attribute>a<key>k<label len>d<string len>L<label><string>
```

attribute	=	0 -	Normal
		1 -	Local display only
		2 -	Transmit only
key	=	1-8	Function key we are setting
label len	=	0-16	Length of the label for the function key. The label has two lines of 8 characters each.
string len	=	-1-80	Number of characters in the function string. A -1 will clear the string.
label	=		The actual function key label string
string	=		The string of characters for the function

Appendix A contains a listing of a udc file that uses function keys in a menu fashion.

III. Slave devices

From the beginning, Hewlett-Packard terminals have supported a wide variety of devices that could be connected off of a terminal. It is hard to believe that one of the first supported serial printers was a 2635 printer slaved off a data capture terminal! These days small printers and plotters are very inexpensive. Providing a quick, easy method to print the screen contents can save a surprising amount of end user and program development time.

The trick to attaching devices to a terminal is matching the type of interface required. Printers can come with RS232 (serial) interfaces, parallel interfaces, HPIB interfaces, or HPIL interfaces. Plotters can have HPIB or RS232 interfaces.

The RS232 (serial) connector on a terminal is a 25 pin D shaped female receptacle. On personal computers it can be a 9 pin male or female D shaped connector. It is usually marked RS232 or serial.

To slave a serial printer to a terminal you need a cable that crosses over pins 2&3. The official Hewlett-Packard cable is 13242G but any cable that you would use to connect a modem to the HP3000 would also work. The pin to pin wiring should be as follows:

TERMINAL / MALE A - B MALE / PRINTER

A	B
1	1
2	3
3	2
4	8
5	20
7	7
8	4
12	19
19	12
20	5

Computer Museum

The HPIB connector is a harmonica style female receptacle that has a cable that bolts on from the side.

The parallel connector is a 25 pin female D receptacle or a harmonica style connector that has clips that hold it on.

The next trick is using the proper cable and getting both the terminal and the slave device configured properly.

Generally the parallel and HPIB devices are easier to cable and configure. The serial devices require a special crossover cable and have many more configuration options.

One large local HP3000 has a large number of HPIB plotters sitting next to Vectra personal computers that can't be used because they don't have a common interface. Also there are many sites with spare PC parallel interface printers that they can't connect to their Hewlett-Packard terminals.

A. HPIB Devices

The original slaved devices were HPIB printers and plotters attached to 264X terminals. The HPIB interface was nice in that multiple devices could be slaved in a daisy chain configuration.

The HP150 also had a built-in HPIB interface and there are many of the 2631 printers with HPIB interfaces floating around because they are no longer supported on the MPE/XL HP3000's

B. Serial Devices

Serial printers and plotters are the most common. To attach a laser printer or desk jet printer to a terminal is relatively simple. The method I use is to configure the printer first and then match the terminal to the printer. The speed, parity, and flow control should be set to 9600 baud, none/8 and xon/xoff. The key to getting the printer working is the proper crossover cable. Attempting to use a straight through cable will not work! In a pinch try using some heavy speaker or telephone wire. Connect pin 2 on the terminal to pin 3 on the printer and pin 3 on the terminal to pin 2 on the printer. Assuming that both the printer and the terminal are both using grounded power cords plugged into a grounded outlet, this should work fine. But you might end up with the nickname of McGyver for a while.

C. Parallel Devices

With the increasing number of personal computers the number of parallel printers has grown and their prices have dropped dramatically. The only problem is that the HP150 and most Hewlett-Packard terminals don't come with a parallel printer interface option. Attempting to attach a parallel printer to a serial port will not result in satisfactory results.

VI. Useful Escape Sequences

At times it is a lot easier to enter escape sequences than to search for the key on the keyboard. Maybe I have this attitude because I use so many different terminals.

A. Home cursor and clear screen

Sometimes we don't want someone to see what we have been working on. A more general reason to clear the screen display is that with 100's of lines of screen memory a great deal of information can be gained simply by rolling down the screen. A still better reason for clearing the screen is to avoid burning in the image of whatever we generally have displayed when we leave for home.

Entering an ESCAPE H which homes the cursor to line one column one followed by an ESCAPE J which clears the screen from the cursor down can be very useful.

Sending an ESCAPE &j@ will remove any function key labels. These also have a tendency to burn into the screen. I also find that the terminal is easier on the eyes if I turn off these labels.

Many sites alter their system message catalog so that the log off message includes these escape sequences so that when the user signs off or is aborted the terminal is cleared.

B. Copy all

On most Hewlett-Packard terminals, to copy the screen contents to the printer you have two choices. You can log data from the bottom of the screen, which sends each line as it is received to the slaved printer. Or, you can position the cursor at the starting point to copy and use the copy all function key. Both have their places. The escape sequence for copy all is ESCAPE 0. The escape sequence for enabling log bottom is much more complex.

One very nice time to use ESCAPE 0 is when you want to get a copy of a block mode screen. Just issue an ESCAPE H ESCAPE 0 and the screen contents will be sent to the slaved printer. This is very useful for documentation and debugging purposes.

Appendix A

Sample UDC File Using Function Keys

This UDC file has evolved over the years and is specifically adapted for an MPE/V development system. The first UDC sets up the initial set of function keys. After the logon UDC is a block of commands that are called from subsequent function key UDC commands.

The actual function key commands consist of a COMMENT line for each of the eight function keys. Each of these lines ends in an <ESC>A which moves the cursor up one line to keep the COMMENT line from rolling down. Note that <ESC> is used to denote the escape character which is a non-displayable character. The final line uses an <ESC>G<ESC>K<ESC>A <ESC>&jB which clears the current and previous line then turns on the display of the function keys. All of this is needed to give the illusion that nothing happened except that the function keys are now set.

The programs and commands enclosed in this UDC file have some interesting entry points and parameters. It is up to the reader to use his or her own judgment as to their usage on a production HP3000.

```
SOFTKEYS
OPTION LOGON,NOLIST
FILE LP;DEV=21
FILE T;DEV=7
FILE TRK;REC=128,1,U,BINARY
MAINMENU
***
E
EDITKEYS
RUN EDITOR.PUB.SYS
OFFIKEYS
***
B
BASKEYS
RUN BASIC.PUB.SYS
PROGKEYS
****
S
SPOOKEYS
RUN SPOOK5.PUB.SYS
MPEVKEYS
***
HPSLATE
RUN HPSLATE.PUB.SYS
OFFIKEYS
*****
HPDRAW
RUN HPDRAW.PUB.SYS
OFFIKEYS
***
HPWORD
RUN HPWORD.PUB.SYS
OFFIKEYS
*****
GRAPH
RUN GRAPH.PUB.SYS
OFFIKEYS
***
CHART
RUN EZCHART.PUB.SYS
OFFIKEYS
***
HPDESK
RUN HPMAIL.HPMAIL.SYS;LIB=G
OFFIKEYS
***
DEBUG
```

```
DEBGKEYS
DEBUG
PROGKEYS
***
DBUTIL
DBUTKEYS
RUN DBUTIL.PUB.SYS
IMAGKEYS
***
MAINMENU
OPTION LIST
COMMENT<ESC>&f2a1k16d08L show    commandsSHOWKEYS<ESC>A
COMMENT<ESC>&f2a2k16d08L listf      keys   LISTKEYS<ESC>A
COMMENT<ESC>&f2a3k16d08L telesup    pub    TELEKEYS<ESC>A
COMMENT<ESC>&f2a4k16d08L telesup    prv    PRIVKEYS<ESC>A
COMMENT<ESC>&f2a5k16d08L  mpe      utils   MPEVKEYS<ESC>A
COMMENT<ESC>&f2a6k16d21L UHAUL     FILES   RUN UHAUL.PUB.TELESUP<ESC>A
COMMENT<ESC>&f2a7k16d08L office     keys   OFFIKEYS<ESC>A
COMMENT<ESC>&f2a8k16d08Lprogram     keys   PROGKEYS<ESC>A
COMMENT<ESC>G<ESC>K<ESC>A<ESC>K<ESC>A<ESC>&jB
***
BASKEYS
OPTION LIST
COMMENT<ESC>&f0a1k16d20LL REC=80  OUT=? LIST,RECSIZE=80,OUT=<ESC>A
COMMENT<ESC>&f0a2k16d27LL
NO,REC=80,OUT=LIST,RECSIZE=80,NONAME,OUT=<ESC>A
COMMENT<ESC>&f2a3k16d04L  LIST             LIST<ESC>A
COMMENT<ESC>&f2a4k16d04L  SAVE             SAVE<ESC>A
COMMENT<ESC>&f2a5k16d09L  SAVE      FAST   SAVE,FAST<ESC>A
COMMENT<ESC>&f0a6k16d06L PURGE      ???    PURGE <ESC>A
COMMENT<ESC>&f0a7k16d07L DELETE     ???    DELETE <ESC>A
COMMENT<ESC>&f2a8k16d04L  EXIT             EXIT<ESC>A
COMMENT<ESC>G<ESC>K<ESC>A<ESC>K<ESC>A<ESC>&jB
***
EDITKEYS
OPTION LIST
COMMENT<ESC>&f2a1k16d20LS LEN=80 RIG=80 S LENGTH=80,RIGHT=80<ESC>A
COMMENT<ESC>&f2a2k16d20LS RIG=80 LEN=80 S RIGHT=80,LENGTH=80<ESC>A
COMMENT<ESC>&f2a3k16d05L L ALL             L ALL<ESC>A
COMMENT<ESC>&f2a4k16d06L LQ ALL            LQ ALL<ESC>A
COMMENT<ESC>&f2a5k16d08L   FQ       FIRST FQ FIRST<ESC>A
COMMENT<ESC>&f2a6k16d09L  LIST    *-5/*+5L *-5/*+5<ESC>A
COMMENT<ESC>&f2a7k16d03L  K;E              K;E<ESC>A
COMMENT<ESC>&f2a8k16d04L  EXIT             EXIT<ESC>A
COMMENT<ESC>G<ESC>K<ESC>A<ESC>K<ESC>A<ESC>&jB
***
TELEKEYS
OPTION LIST
COMMENT<ESC>&f2a1k16d24LBULDACCT           RUN BULDACCT.PUB.TELESUP<ESC>A
COMMENT<ESC>&f2a2k16d24LDBLOADNG           RUN DBLOADNG.PUB.TELESUP<ESC>A
COMMENT<ESC>&f2a3k16d24LEPTFIND            RUN EPTFIND.PUB.TELESUP <ESC>A
COMMENT<ESC>&f2a4k16d24L  LIST             RUN LIST.PUB.TELESUP    <ESC>A
COMMENT<ESC>&f2a5k16d24LPSCREEN            RUN PSCREEN.PUB.TELESUP <ESC>A
COMMENT<ESC>&f2a6k16d24LUDCWHO             RUN UDCWHO.PUB.TELESUP  <ESC>A
COMMENT<ESC>&f2a7k16d24LLOGAUDIT           RUN LOGAUDIT.PUB.TELESUP<ESC>A
COMMENT<ESC>&f2a8k16d08LPREVIOUS  MENU  MAINMENU<ESC>A
COMMENT<ESC>G<ESC>K<ESC>A<ESC>K<ESC>A<ESC>&jB
***
PRIVKEYS
OPTION LIST
COMMENT<ESC>&f2a1k16d24L DIRPUR            RUN DIRPUR.PRV.TELESUP  <ESC>A
COMMENT<ESC>&f2a2k16d24L  IDAT             RUN IDAT.PRV.TELESUP    <ESC>A
COMMENT<ESC>&f2a3k16d24LFLUSHER5           RUN FLUSHER5.PRV.TELESUP<ESC>A
COMMENT<ESC>&f2a4k16d24LPROGINFO           RUN PROGINFO.PRV.TELESUP<ESC>A
COMMENT<ESC>&f2a5k16d24LPROGSIZE           RUN PROGSIZE.PRV.TELESUP<ESC>A
COMMENT<ESC>&f2a6k16d24LSYSINFO            RUN SYSINFO.PRV.TELESUP <ESC>A
COMMENT<ESC>&f2a7k16d24LTAPELIST           RUN TAPLIS.PRV.TELESUP  <ESC>A
COMMENT<ESC>&f2a8k16d08LPREVIOUS  MENU  MAINMENU<ESC>A
COMMENT<ESC>G<ESC>K<ESC>A<ESC>K<ESC>A<ESC>&jB
***
LISTKEYS
OPTION LIST
COMMENT<ESC>&f2a1k16d05L LISTF             LISTF<ESC>A
COMMENT<ESC>&f2a2k16d07L LISTF      @,1    LISTF,1<ESC>A
COMMENT<ESC>&f2a3k16d07L LISTF      @,2    LISTF,2<ESC>A
```

```
COMMENT<ESC>&f2a4k16d13L   L2    @UDC@  LISTF @UDC@,2<ESC>A
COMMENT<ESC>&f2a5k16d21L  PRV   TELESUP LISTF @.PRV.TELESUP,2<ESC>A
COMMENT<ESC>&f2a6k16d21L  PUB   TELESUP LISTF @.PUB.TELESUP,2<ESC>A
COMMENT<ESC>&f2a7k16d08LPREVIOUS  MENU  MAINMENU<ESC>A
COMMENT<ESC>&f2a8k16d08LSHOWTIME        SHOWTIME<ESC>A
COMMENT<ESC>G<ESC>K<ESC>A<ESC>K<ESC>A<ESC>&jB
***
SHOWKEYS
OPTION LIST
COMMENT<ESC>&f2a1k16d07LSHOWJOB         SHOWJOB<ESC>A
COMMENT<ESC>&f2a2k16d16LSHOWOUT SP;JOB=@SHOWOUT SP;JOB=@<ESC>A
COMMENT<ESC>&f2a3k16d15L  SHOW  SESSIONSSHOWJOB ;JOB=@S<ESC>A
COMMENT<ESC>&f2a4k16d15L  SHOW    JOBS  SHOWJOB ;JOB=@J<ESC>A
COMMENT<ESC>&f2a5k16d07LSHOWDEV         SHOWDEV<ESC>A
COMMENT<ESC>&f2a6k16d05L SHOWQ          SHOWQ<ESC>A
COMMENT<ESC>&f2a7k16d09L  SHOW    CACHE SHOWCACHE<ESC>A
COMMENT<ESC>&f2a8k16d08LPREVIOUS  MENU  MAINMENU<ESC>A
COMMENT<ESC>G<ESC>K<ESC>A<ESC>K<ESC>A<ESC>&jB
***
SPOOKEYS
OPTION LIST
COMMENT<ESC>&f0a1k16d05L  TEXT    ???   TEXT <ESC>A
COMMENT<ESC>&f2a2k16d08L  SHOW    @.@   SHOW @.@<ESC>A
COMMENT<ESC>&f2a3k16d10L  SHOW   @.@;@  SHOW @.@;@<ESC>A
COMMENT<ESC>&f2a4k16d08L  LIST    ALL   LIST ALL<ESC>A
COMMENT<ESC>&f0a5k16d05L  FIND    ???   FIND <ESC>A
COMMENT<ESC>&f0a6k16d06L PURGE    ???   PURGE <ESC>A
COMMENT<ESC>&f2a7k16d06L XPLAIN         XPLAIN<ESC>A
COMMENT<ESC>&f2a8k16d04L  EXIT          EXIT<ESC>A
COMMENT<ESC>G<ESC>K<ESC>A<ESC>K<ESC>A<ESC>&jB
***
MPEVKEYS
OPTION LIST
COMMENT<ESC>&f2a1k16d17L FREE5          RUN FREE5.PUB.SYS<ESC>A
COMMENT<ESC>&f2a2k16d20LLISTDIR5        RUN LISTDIR5.PUB.SYS<ESC>A
COMMENT<ESC>&f2a3k16d01L SPOOK          S<ESC>A
COMMENT<ESC>&f2a4k16d17L  OPT     PROG  RUN   OPT.PUB.SYS<ESC>A
COMMENT<ESC>&f2a5k16d08L image   utils  IMAGKEYS<ESC>A
COMMENT<ESC>&f2a6k16d19LTERMDSM         RUN TERMDSM.PUB.SYS<ESC>A
COMMENT<ESC>&f2a7k16d08LPREVIOUS  MENU  MAINMENU<ESC>A
COMMENT<ESC>&f2a8k16d04L  EXIT          EXIT<ESC>A
COMMENT<ESC>G<ESC>K<ESC>A<ESC>K<ESC>A<ESC>&jB
***
OFFIKEYS
OPTION LIST
COMMENT<ESC>&f2a1k16d06L HPWORD         HPWORD<ESC>A
COMMENT<ESC>&f2a2k16d07L HPSLATE        HPSLATE<ESC>A
COMMENT<ESC>&f2a3k16d01L editor         E<ESC>A
COMMENT<ESC>&f2a4k16d05L  DSG           GRAPH<ESC>A
COMMENT<ESC>&f2a5k16d05L EZCHART        CHART<ESC>A
COMMENT<ESC>&f2a6k16d06L  HPDRAW        HPDRAW<ESC>A
COMMENT<ESC>&f2a7k16d06L  HPDESK        HPDESK<ESC>A
COMMENT<ESC>&f2a8k16d08LPREVIOUS  MENU  MAINMENU<ESC>A
COMMENT<ESC>G<ESC>K<ESC>A<ESC>K<ESC>A<ESC>&jB
***
PROGKEYS
OPTION LIST
COMMENT<ESC>&f2a1k16d02L  basic         B <ESC>A
COMMENT<ESC>&f2a2k16d19L  TOOL    SET   RUN TOOLSET.PUB.SYS<ESC>A
COMMENT<ESC>&f2a3k16d05L  debug         DEBUG<ESC>A
COMMENT<ESC>&f2a4k16d20L  BASIC   COMP  RUN BASICOMP.PUB.SYS<ESC>A
COMMENT<ESC>&f2a5k16d22L  PREP          PREP $OLDPASS,$NEWPASS<ESC>A
COMMENT<ESC>&f2a6k16d09L  SEG-   MENTER SEGMENTER<ESC>A
COMMENT<ESC>&f2a7k16d04L   EXIT         EXIT<ESC>A
COMMENT<ESC>&f2a8k16d08LPREVIOUS  MENU  MAINMENU<ESC>A
COMMENT<ESC>G<ESC>K<ESC>A<ESC>K<ESC>A<ESC>&jB
***
DEBGKEYS
OPTION LIST
COMMENT<ESC>&f2a1k16d19L  GET     CAPS  MDL-'DL-1'+2       <ESC>A
COMMENT<ESC>&f2a2k16d19L  LIST    CODE  D P,10,C           <ESC>A
COMMENT<ESC>&f2a3k16d19LDISPLAY REGISTERDR                 <ESC>A
COMMENT<ESC>&f2a4k16d19LDISPLAY   FLM   DA,10              <ESC>A
COMMENT<ESC>&f2a5k16d19L   GO           R                  <ESC>A
COMMENT<ESC>&f2a6k16d19LDISPLAY    DB   D,10               <ESC>A
```

```
COMMENT<ESC>&f2a7k16d19L   EXIT          E                          <ESC>A
COMMENT<ESC>&f2a8k16d08LPREVIOUS  MENU  MAINMENU<ESC>A
COMMENT<ESC>G<ESC>K<ESC>A<ESC>K<ESC>A<ESC>&jB
***
IMAGKEYS
OPTION LIST
COMMENT<ESC>&f2a1k16d20LDBSCHEMA        RUN DBSCHEMA.PUB.SYS<ESC>A
COMMENT<ESC>&f2a2k16d24L DBLOAD         RUN DBLOAD.PUB.SYS,DEFER<ESC>A
COMMENT<ESC>&f2a3k16d19L DBRECOV        RUN DBRECOV.PUB.SYS<ESC>A
COMMENT<ESC>&f2a4k16d20LDBUNLOAD        RUN DBUNLOAD.PUB.SYS<ESC>A
COMMENT<ESC>&f2a5k16d06L  dbutil        DBUTIL<ESC>A
COMMENT<ESC>&f2a6k16d17L  QUERY         RUN QUERY.PUB.SYS<ESC>A
COMMENT<ESC>&f2a7k16d20LDBDRIVER        RUN DBDRIVER.PUB.SYS<ESC>A
COMMENT<ESC>&f2a8k16d08LPREVIOUS  MENU  MAINMENU<ESC>A
COMMENT<ESC>G<ESC>K<ESC>A<ESC>K<ESC>A<ESC>&jB
***
DBUTKEYS
OPTION LIST
COMMENT<ESC>&f0a1k16d04L  SET      ???   SET <ESC>A
COMMENT<ESC>&f0a2k16d05L  SHOW     ???   SHOW <ESC>A
COMMENT<ESC>&f0a3k16d07L ENABLE    ???   ENABLE <ESC>A
COMMENT<ESC>&f0a4k16d08LDISABLE    ???   DISABLE <ESC>A
COMMENT<ESC>&f0a5k16d07L CREATE    ???   CREATE <ESC>A
COMMENT<ESC>&f0a6k16d06L PURGE     ???   PURGE <ESC>A
COMMENT<ESC>&f0a7k16d07L ERASE     ???   ERASE  <ESC>A
COMMENT<ESC>&f2a8k16d04L  EXIT           EXIT<ESC>A
COMMENT<ESC>G<ESC>K<ESC>A<ESC>K<ESC>A<ESC>&jB
***
```

Matching Network Servers to User Needs

John Green
Hewlett-Packard
8010 Foothills Blvd.
Roseville, CA 95678

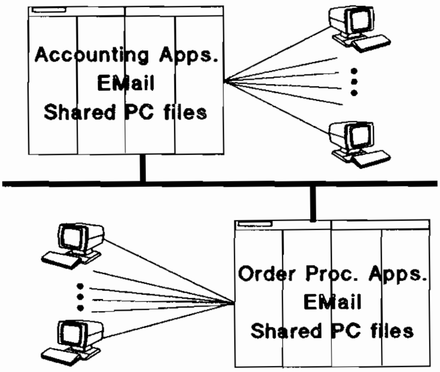
Abstract
Now that industry standards such as LAN Manager are spanning a variety of hardware and software platforms, network designers have more freedom of choice. This paper examines file and print server platforms for client PCs in an office environment from the perspective of moving the PC LAN administrative burden from end-user departments to MIS organizations. First, the paper examines why such a move is desirable and how changes to the LAN topology may reduce the administrative burden. Then, it provides some guidelines for selecting file and print servers based on characteristics of the application(s) being used. The interoperability of server platforms provides the flexibility to mix and match different servers to specific applications. Making this correlation between servers and user needs is a key part of maximizing the overall effectiveness of the network.

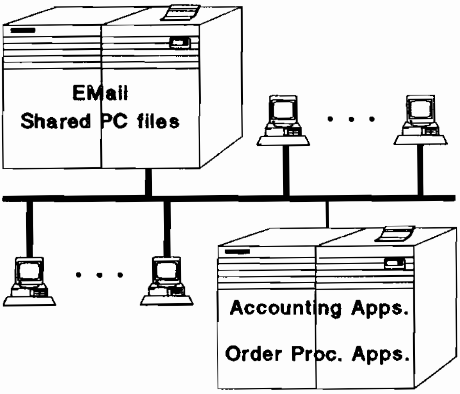
Organizational Evolution
The emergence of PC LANs is enabling gains in workgroup productivity. To fully realize these gains, MIS departments must play a larger role in PC LAN implementations. In many organizations today, PC LANs are limited to those departments with the expertise and resources to manage the LAN internally. Departments without the expertise or resources are unable to exploit PC LAN technology by themselves.

The decentralized management of PC LANs is inefficient. The adequacy of resource management procedures, such as backing up and controlling access to sensitive data, is often questionable and almost always less stringent than procedures used in MIS environments. Expertise is spread among several individuals who administer the LAN as a part-time job. In addition to higher training costs, opportunities to consolidate and streamline operations are often lost. By one estimation, a departmental approach to PC LAN administration requires the equivalent of one full time administrator per 100 end-users.

As MIS organizations assume a larger role in PC LANs, changes to administration procedures will be necessary since fiscal constraints rule out adding MIS employees at the same rate they are needed in a departmental approach (one full-time MIS employee per 100 end-users). Consolidation of processing functions onto fewer larger machines will be necessary.

New Topologies
LAN topologies allow for more consolidation and specialization than traditional RS-232 terminal connections. In a host/terminal environment, processing is partitioned by user groups or departments. Each user typically accesses a single machine for most, if not all, of their processing needs. Although each user group has a unique set of processing needs, there is often a common set of functionality among the various departments. For example, while the Accounting department needs access to accounting applications and the Order Processing departments must access order-processing applications, both may require access to electronic mail and word processing applications. Due to the nature of RS-232 connections it is simpler to duplicate common functionality on multiple machines than it is to wire a single terminal to several machines. Even with electronic switching equipment such as a PBX or DTC, the terminal is only connected to a single host machine at a time.





In a LAN environment, the connectivity issues are largely removed allowing functionality which is common between various user groups to be consolidated onto fewer machines.

Electronic Mail is a prime candidate for consolidation. By consolidating mail nodes, you save disk space, eliminate processing associated with internode message traffic, and reduce administration costs.

Similar savings can be realized through the consolidation of file and print sharing. Instead of administering several micro-computer (i.e. Intel 386 based) servers, a mini-computer can be used in their place.

In addition to reducing administration costs, mini-computers provide other benefits as well. Security may be enhanced since micro-computers are most often found in the users' workplace while mini-computers tend to reside behind the closed doors of the computer room. Mini-computers have a broader family of models with a clear growth path to meet future needs. Mini-computers are bigger. They support larger disk, memory, and peripheral (printer/plotter) configurations than micro-computers. Mini-computers have high availability features such as disk-mirroring, online backup, auto-restart, etc. And, of course, mini-computers are already under control of the MIS department with established operating procedures for backup, security, etc.

Mini-computers should not be treated as a panacea, however, since there are situations where micro-computers do a better job. Distance limitations associated with connecting a printer in the work area to the mini-computer in the data center can be an issue in some situations. Micro-computer servers can reside in the users work area eliminating any distance limitations associated with connecting peripherals. Consolidating several user groups onto a machine can expose the large user community of the mini-computer to performance degradations stemming from a particularly intense workload generated by a small group of users. Micro-computers servers are relatively inexpensive compared to mini-computer servers and can be used to offload intense workloads that would otherwise impact the entire community of mini-computer users. Therefore, the best strategy is to consolidate as much as possible onto mini-computers and use micro-computers in those situations where they provide a superior solution. Understanding the PC application workload is a necessary prerequisite to choosing a file and print server.

Application Workloads

As always, it is essential to consider the workload users will place on your systems. In the case of file and print sharing, this means looking at the PC applications and how they will interact with the server. When the application software is installed on the local hard-disk of the PC and shared disks are only used for data, there are "typically" (application dependent) three basic types of transactions.

These are:

File Load - Copying file from the server to PC memory.
File Save - Copying file from PC memory to the server.
File Print - Copying file from PC memory to a spooler on the server.

WordPerfect and MS-Word are "typical" applications. When you select a document for editing, a File Load transaction occurs. As you edit the document, all your processing takes place locally on the PC. When you save the document, a File Save transaction occurs. Not all applications are "typical".

Some applications, such as those in the NewWave environment tend to use shared disks less frequently than "typical" applications. NewWave workstations are a good example of this. The Object Management Facility (OMF) of NewWave must reside on the local hard disk. To remain under control of the OMF, data files must also reside on the local hard disk (i.e. no distributed objects - yet). NewWave has an object storage facility that permits objects to be exported to a shared disk where they can be imported by other users. Moving objects to/from the shared disk consumes slightly more server resources than a simple File Load or File Save because a second directory file is updated and a connection to the shared disk may need to be established as part of the transaction. Object storage transfers should be less frequent than File Loads and File Saves from a "typical" application. Therefore, NewWave workstations are likely to generate a less intense workload than "typical" applications.

Other applications, such as PC databases, may access several files simultaneously and perform random I/O as opposed to the sequential I/O used in "typical" transactions. For example, adding a record to a DBASE database that uses many indexes, may involve random I/O to several files. PC Databases tend to generate a much more intense workload than "typical" applications. Graphics printing may also consume a significant amount of processing. Graphics printing to a LaserJet involves sending raster images to the print server. These images can be rather large and if many images are generated, they could create a noticeable impact on the server.

Running programs from a shared disk can also generate an intense workload although it depends so much on the particular application that it is difficult to make any general characterizations. Many word processing applications will frequently post modifications of a document to a temporary file to avoid losing work in the event of a PC failure. It is not uncommon for temporary files to reside in the subdirectory where the PC application is installed. As you move around (change screens, etc.) within the application, portions of code may be copied from the shared disk to PC memory. With MS-DOS, a common way to determine if a file exists is to try to open it. Some applications will open several files during initialization. For example, a word processor might open all its font files during initialization to determine which fonts exits.

File sharing has proven to be the "first step" application used by PCs over a LAN because it is a simple, transparent extension of PC resources. Most of today's PC applications have all their intelligence implemented in the client because they were developed as single-user applications. Even the multi-user applications available today are single-user applications that have been modified to accommodate access from multiple users. In this environment, the server processes primitive requests. The performance/capacity of the server is largely determined by its ability to transmit data across the LAN. That is, the requests are so primitive that the majority of the processing involves receiving the request and sending the reply. Actual processing to generate the reply is small by comparison.

As client/server applications emerge, the server software will evolve from a process that knows nothing about the application to an intelligent back-end of the application. For example, adding a record to a database will involve a single SQL request instead of a series of primitive file sharing requests. The performance/capacity of the server will largely be determined by its ability to process these sophisticated requests rather than its ability to simply send and receive data.

Now that we have a basic understanding of the different types of workloads that can be generated by PC applications, let's look at how the various server platforms accommodate these workloads.

HP 3000

The HP 3000 is highly tuned for On-Line Transaction Processing (OLTP). The price/performance advantages of MPE XL for OLTP applications tend to make it the information resource for the office. NewWave Office Information Access Services allow PC users to exploit these information resources to their fullest advantage.

HP DeskManager combined with the wide area networking capabilities of the HP 3000 make an excellent solution for an electronic mail server.

Shared disks for PC clients installed on the HP 3000 provide a good foundation for integrating the MPE and MS-DOS environments. All MS-DOS processing must be initiated from the PC due to architectural limitations of MS-DOS. This makes it impossible for an MPE user to initiate file transfers to and from DOS clients. However, with shared disks on the HP 3000, it is possible for an MPE user to transfer files between the MPE and MS-DOS domains opening the door for many forms of MPE/DOS integration that is transparent to the PC user. For example, a report generated by an MPE application can be "pushed" onto a shared disks where PC users can access it from their word processing applications. All the PC users need to know is the name of the shared disk resource and the MS-DOS filename (including path). They do not have to know anything about MPE.

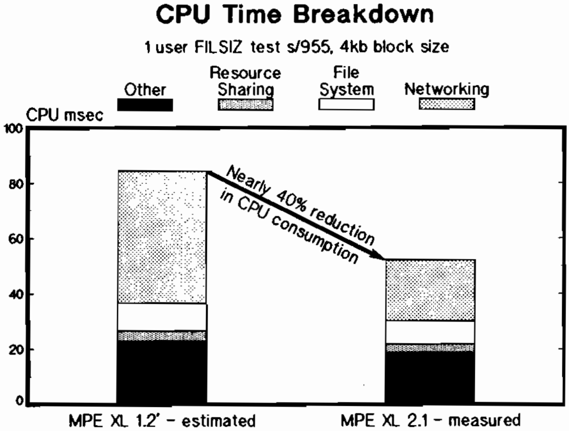
The workload characteristics of OLTP applications are different than disk file sharing with PC clients. For example, OLTP applications tend to work with a finite set of files and transactions. The ratio of file READs and WRITEs to file OPENs and CLOSEs tends to be very high. A database consisting of a few dozen files may remain open all day long as OLTP applications READ and WRITE to it.

By contrast, a word processing application tends to work with many different document files throughout the day. A document file may be opened and read into the PC client's memory, edited in the PC client's memory, and then written back to the HP 3000 disk before it is closed. If the file is small, such as a one or two page memo, it may be read or written in a single request. This type of processing results in a very low ratio of READs/WRITEs to OPENs/CLOSEs.

Other differences exist in the file sizes and directory structures. MPE files tend to be much larger than MS-DOS files. MPE organizes files into groups and accounts while MS-DOS uses a tree hierarchy of directories and subdirectories.

In spite of these differences, MPE performs reasonably well as a file and print server. If you compare the HP 3000 to an IBM AS/400 you will very likely find the HP 3000 has superior file and print sharing performance. This is an "apples-to-apples" comparison that I would encourage you to make. However, if you compare an HP 3000 with a Vectra/386 running Novell Netware, you are likely to find that Novell has superior file sharing performance while the HP 3000 has superior print sharing performance. As previously discussed, there are many inherent benefits to using mini-computers over micro-computers and a comparison between an HP 3000 and a Vectra/386 should not be made on performance alone.

All the workload differences between OLTP applications and PC file sharing that I have mentioned are related to the file systems. Yet, when you look at the total amount of processing that takes place on a file and print server, only a small percentage of time is spent in the file system. The following BAR charts illustrate this. On MPE XL 2.1 the time spent in networking software was nearly cut in half from what it was on MPE XL 1.2. This resulted in a 40% reduction in total CPU consumption. This particular improvement is mainly the result of moving NetIPC and IOWAIT to Native Mode. Continued improvements to MPE XL will be made to benefit both OLTP and file sharing workloads.



Although file system differences should not be over emphasized they do suggest the HP 3000 is most effectively utilized when file and print sharing is a secondary application. In other words, the HP 3000 should primarily be used for an application that exploits its commercial data processing capabilities. Additional HP 3000 capacity can be used for file and print sharing. These file system differences also suggest that the HP 3000 is vulnerable to intense file sharing workloads. While the HP 3000 performs quite satisfactorily with "typical" applications such as WordPerfect or low intensity applications such as NewWave, you should be cautious about putting an intense file sharing application on the HP 3000.

The current MPE product (Resource Sharing) is implemented on top of the MPE file system. Every MS-DOS file on the shared disk maps to a corresponding MPE file. LAN Manager has recently emerged as an industry standard and HP is currently developing LAN Manager for MPE XL. Many implementation decisions have yet to be made but it is possible that LAN Manager for MPE XL will be implemented differently than the current Resource Sharing product. Regardless of how its implemented, LAN Manager will allow application developers to build client/server applications that are network independent. As already mentioned, PC databases present an intense workload because of the primitive nature of the file sharing requests. As these applications evolve to a client/server model, they may use more sophisticated requests which more fully exploit high performance OLTP systems.

To summarize, in addition to providing all the advantages of a mini-computer, the HP 3000 provides exceptional OLTP performance. As we look for ways to reduce administration costs by consolidating functionality onto fewer larger machines, the HP 3000 is an excellent place to consolidate Electronic Mail or to consolidate file and print sharing along with another function that more fully exploits the virtues of MPE. Combining file and print serving with commercial data processing opens the door for many forms of PC integration. Other platforms should be considered for intense file sharing workloads.

HP 9000

The HP 9000 is a general purpose system based on standards. The versatility of the HP 9000 allow it to be used for a wide variety of applications including data processing. NewWave Office for HP-UX brings much of the office functionality which was formerly only found on MPE to HP-UX.

Since it is used for a wide variety of applications, the operating system (HP-UX) is "lean and mean" and not specialized for a particular type of processing.

The HP-UX and MS-DOS file systems closely resemble each other. They both organize files into a tree structure of directories and subdirectories. They are both primitive in comparison to MPE (e.g. no transaction management, no fixed length records - just strings of bytes, etc.). These similarities allow HP LAN Manager/X to efficiently process the primitive file request from the MS-DOS clients.

The shared resource services of NewWave Office for HP-UX are provided HP LAN Manager/X. HP LAN Manager/X is based on the OS/2 LAN Manager standard developed by MicroSoft.

LAN Manager is a superset of MS-NET. This provides compatibility with OfficeShare (MS-NET) clients. An OfficeShare client can communicate with LAN Manager servers and LAN Manager clients can communicate with MS-NET servers such as Resource Sharing on the HP 3000. LAN Manager provides the following features in addition to MS-NET file and printer sharing.

1) Enhanced Remote File Access (RFA) protocol for improved performance. These enhancements allow requests to be chained together. For example, instead of sending a LOCK request followed by a READ request. The client can send a single LOCK & READ request.

2) User-Level security. LAN Manager has two security modes. Share-Level security is the same as MS-NET security. Passwords are assigned to resources and a user must know the password to gain access to the resource. With User-Level security, access to resources is

controlled by user identification. Instead of configuring a password, the administrator maintains a list of users (or groups of users) permitted to access the resource.

3) Network independent Application Program Interfaces (API). The MS-NET standard did not include APIs. Application developers were limited to proprietary interfaces such as NetIPC.

   In addition to Named Pipes and MailSlots which are the APIs for program-to-program communication, LAN Manager defines an Administrative API which allows for remote administration of a server.

The HP LAN Manager/X implementation on HP 9000 computers does not currently include the full set of ADMIN APIs. However, remote administration is possible using virtual terminal services.

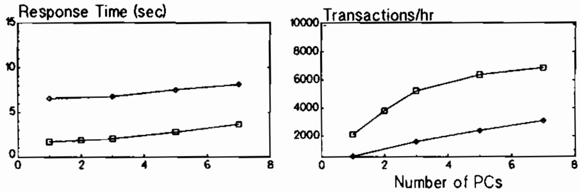
HP LAN Manager/X performance is competitive with Novell Netware/386. In many instances, HP LAN Manager/X running on an HP 9000 has more capacity than Novell Netware/386 running on the highest performance Vectra/386. The following graphs compare an HP 9000 model 835 with a Vectra RS25C.

Looking at the three "typical" transactions performed by PC Applications we see that FILE LOAD performance is nearly the same for both products. HP LAN Manager/X has more capacity for FILE SAVES than Novell Netware/386. When the server is not heavily loaded, Novell Netware/386 performs FILE SAVES faster than HP LAN Manager/X. However, as the number of client PCs increases, HP LAN Manager/X maintains its FILE SAVE performance while Novell Netware degrades. Adding disk drives to the Novell system did not improve the multi-user FILE SAVE performance. HP LAN Manager is faster than Novell Netware/386 at FILE PRINTs.

Although HP LAN Manager/X is faster with sequential file access, Novell Netware/386 has advantages with random file I/O. This is revealed by the BENCH23 simulation testing. PC database applications tend to use random file I/O.

FILSIZ Benchmark

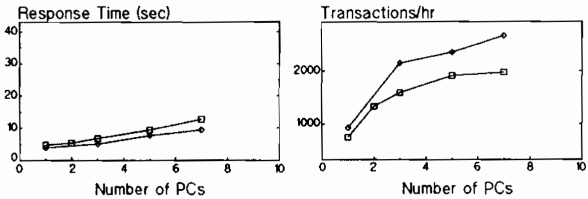
HP LAN ManagerX/835 —□—   Novell Netware 386/RS25C —◇—



FILSIZ: Randomized mix (per PC) of 60% reads; 20% writes; 20% prints.
File size ranges from 1Kb through 1Mb
A PCBENCH1 benchmark.

READ1MB Benchmark

HP LAN ManagerX/835 —□—   Novell Netware 386/RS25C —◇—

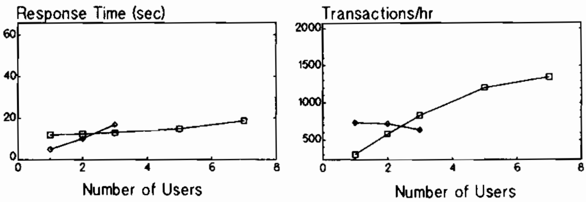


READ1MB: Each user performs 20 1MB reads from server to memory.

A PCBENCH1 benchmark.

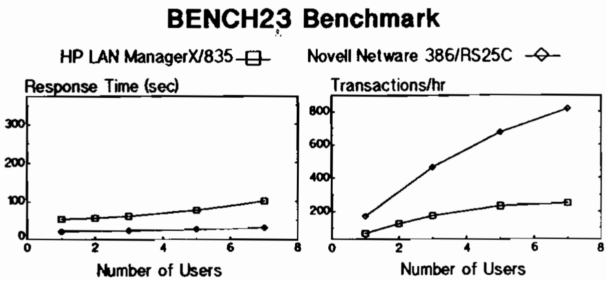
WRITE1MB Benchmark

HP LAN ManagerX/835 —□—   Novell Netware 386/RS25C —◇—



WRITE1MB: Each user performs 20 1MB writes from memory to server.

A PCBENCH1 benchmark.



BENCH23: A simulation of PC Magazine's BENCH23. Performs sequential and random I/O to a 256 Kbyte file with block sizes of 512 and 4096 bytes. A PCBENCH1 script.

Once again, it is important to understand the PC applications before drawing any conclusions. Are your users using "typical" applications as already described or are they using an application such as a PC database? How do the applications print? Some applications will not realize a difference in printing performance. For example, Drawing Gallery does a significant amount of processing as it generates output for a printer or plotter. In this case, the PC is not sending data to the server at a fast enough rate for any difference in print performance to be realized.

To summarize, the HP 9000 provides all the advantages of a mini-computer with excellent file and print sharing performance. The generic properties of HP-UX make the HP 9000 extremely versatile. Like the HP 3000, the HP 9000 can be used for both commercial data processing applications and file and print sharing services. However, the HP 9000 is less vulnerable to intense file sharing workloads since it can deliver performance that is similar to what intense file sharing users have become accustom to with their micro-computer solutions. However, you still need to examine your PC application workloads to ensure no small group of users will "hog" the machine with an intense application.

As we look for ways to reduce administration costs through consolidation of functionality, the HP 9000 could be used as a high capacity file server and/or electronic mail server. Just as HP DeskManager combined with the wide area networking capabilities of the HP 3000 make for an excellent mail server, HP OpenMail can be combined with the wide area networking capabilities of the HP 9000 to make an equally outstanding mail server.

OS/2 386/486

OS/2 LAN Manager includes implementation of the ADMIN API. This permits someone in the MIS department to remotely administer OS/2 servers out in departmental work areas. All OS/2 servers on the LAN can be administered and monitored from a single OS/2 workstation. No additional networking products are needed for remote administration.

As already mentioned, micro-computers should be used for for offloading intense workloads that would otherwise impact the entire community of mini-computer users and/or for remote printing. Backing up remote OS/2 servers can create an administrative burden. However, if the intense workload results from graphics printing or running applications from a shared disk, backups should not be required on a regular basis. If the intense workload results from PC database applications, the particular files requiring backup can be copied to a mini-computer shared disk on a regular basis.

As we look for ways to reduce administration cost through consolidation of functionality, the remote administration capabilities of OS/2 LAN Manager servers provide an excellent platform for handling our remote printing needs or for offloading an intense application from the mini-computer.

Although MPE XL, HP-UX, and OS/2 all have different administrative interfaces today, this should change as HP standardizes on LAN Manager and implements the ADMIN APIs. This suggests that training investments to develop expertise in OS/2 LAN Manager administration will be protected.

Novell Netware

The decentralized departmental approach that has characterized much of PC LAN computing to date has led to decentralized purchase decisions. Although LAN Manager has emerged as a standard that should be considered for future purchases, many departments already have Novell solutions installed.

An HP NS LAN Gateway can be used to connect Novell users to an HP 3000. Just as the HP 3000 is highly tuned for OLTP, Novell servers are highly tuned for file sharing (printer sharing may be another matter) and do an exceptionally good job with applications involving random I/O. Using a gateway to connect the two environments allows you to exploit the strengths of both.

Like OS/2 LAN Manager servers, Novell servers can be used to isolate the large user community of the mini-computer from particularly intense workloads generated by a small group of users.

If the intense workload results from running applications off a shared disk, backups should not be required on a regular basis. If the intense workload is the result of PC database applications, the number of data files may be small enough (less that 50) to allow them to be copied to an HP 3000 shared disk on a regular basis. If the number of files requiring backup is large (more than 1000), the performance of copying files through the HP NS LAN Gateway to the HP 3000 may prohibit this type of approach.

Conclusions

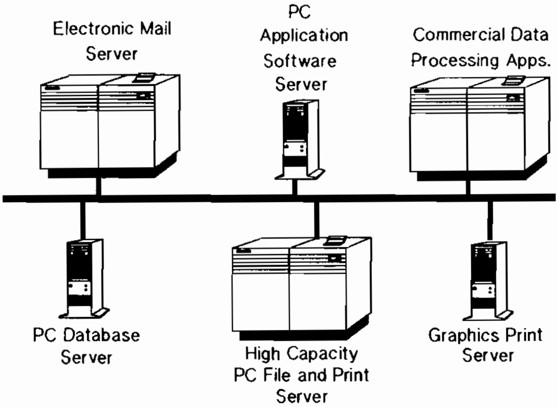
To fully reap the benefits of PC LAN technology, MIS departments will play a larger role in PC LAN administration. As MIS departments assume this larger role, it will be necessary to consolidate functionality onto fewer machines to achieve better economies of scale. LAN topologies allow clients PCs to be connected to several servers simultaneously permitting common functionality to be consolidated.

Mini-computers are a logical solution for consolidating functionality. They have well-established operational procedures for backing up data and managing system resources. They also support large hardware configurations.

Micro-computers can supplement mini-computers by providing excellent price/performance for handling intense file sharing workloads or for printing in a remote workarea.

Understanding PC application workloads is an absolute prerequisite to consolidating file and printer sharing functionality. Since many PC LANs have been implemented departmentally, MIS organizations are often unaware of how the server is actually used. Consolidating file and printer sharing without knowledge of the workload can be very risky since large user communities may be impacted by an intense workload from a small group of users.

The interoperability of file and print serving platforms allow you to balance your consolidation objectives against the diverse performance needs of your users. HP 3000s cannot be beat for OLTP applications. They should be used primarily for these types of applications. Adding PC file sharing to an HP 3000 opens the door for many forms of office/data-processing integration. HP 9000s are very versatile. The HP 9000 has all the advantages of a mini-computer and can deliver PC file sharing performance that is competitive with the leading micro-computer solution.



Matching Network Servers 7083-14

NEWWAVE OFFICE BENCHMARKING - Appendix A

INTRODUCTION

This article provides an overview of the performance benchmarking techniques used. These techniques are used in the presentation of performance information comparing HP solutions to other HP solutions, as well as comparing HP solutions to those of competitors.

OBJECTIVE

This article presents an analysis of our benchmarking process as well as explaining the benchmarks used. This article should be used as a reference when applying information from other articles presented by Data Management Systems Division (DMSD).

APPLICATION DESCRIPTION

We created a simple and transportable environment in which we can execute scripts against any server. This environment is built around a program developed by DMSD called PCBench1.

Blocking size refers to the amount of data sent to the networking software in a single call. Varying the blocking size demonstrates the efficiency of the networking code and allows us to simulate actual applications.

PCBench1 runs on an MS-DOS client that can be networked to any server under test. Either interactively, or in batch, PCBench1 executes specific commands to perform a file read, file write, file print, execute a MS-DOS command, or delay to simulate user think time. With near microsecond accuracy, the response time for these transactions is obtained. By scripting a number of these commands, and controlling the blocking of I/O and using appropriate delays, the network activity created by most applications can be generated.

Internally, PCBench1 performs I/O at block sizes up to 64Kb, limited by MS-DOS. The file open and close time is reported as well as the data transfer time for file I/O. All I/O is to or from memory, such that local hard disk performance is not a variable in the test environment.

TEST METHODOLOGY

In batch mode, PCBench1 is run redirecting the input from a script and redirecting the output to a file. A reduction process enacted on the output file produces statistical results including average transaction response time, standard deviation, transactions per hour, as well as summary data. Server resource utilization is monitored where possible but not all operating systems have performance monitoring or collection tools available.

Benchmark Scripts

We have developed several benchmark scripts to analyze different areas of the server. To avoid synchronization of PC activity, some scripts have been randomized so different transactions are running on each client PC at any given time. Each PC connects to its own shared disk on the server to alleviate share violations. The benchmark descriptions are as follows:

FILSIZ - This script performs reads and writes from 1Kb through 1Mb, and prints from 1Kb through 512Kb, all in logical increments executed in random order. About 60% of this script's transactions are reads, 20% are writes and 20% are file prints. The block size for all reads and writes is 64Kb, or the file size which ever is smaller. This script has randomized versions to avoid synchronization.

READ1MB - This script performs 20 reads of a 1Mb file, in 64Kb blocks.

WRITE1MB - This script performs 20 writes of a 1Mb file, in 64Kb blocks.

PRINT1MB - This script performs six prints of a 1Mb file, in 64Kb blocks.

BENCH23 - This is a script modeled after PC Magazine's BENCH23 benchmark. It performs sequential and random reads and writes of a 256Kb file with a block sizes of 512b and 4Kb. This benchmark is run to show the random I/O portion of the script.

Most of the benchmarks described above do not simulate applications. Each application acts differently from the network's perspective, so rather than simulate many applications we look at how the server reacts to a variety of file sizes and blocking sizes.

You will notice that none of the industry standard benchmarks are included in this set of benchmarks (except for the Bench23 benchmark). In analyzing the industry standard benchmarks, none of them were able to simulate the workload of real users running applications. The industry standard benchmarks do not allow the flexibility to vary the blocking sizes and file sizes seen by the network from each application. PCBench1 allows all the flexibility anyone would need to simulate almost any PC application. PCBench1 can also simulate industry standard file and print benchmarks.

The output of the testing produces transactions per hour and response time. As the number of clients increase, the system bottlenecks can be identified for each workload scenario. In this way the workload of hundreds of users can be put on a server using only a few PCs.

All of our tests are made on systems with file/print sharing as the only workload, but your system will probably have other workloads on it as well. This point must always be kept in mind.

Now let's take a critical look at the drawbacks of the tool. It would be nice if we could say each PC acted exactly like many actual users, but it doesn't. A single PC only has one process on the server, so there is a substantial difference in the amount of memory consumed. With only a few PCs, there is less contention between processes than if there were actually several hundred processes contending for the CPU. On systems with large amounts of cache memory, a file that is read for the second time may be in memory whereas an actual user may require disk I/Os. This testing tool presents server performance at its best. Further testing like the max connection tests validate our final performance conclusions.

TEST ENVIRONMENT

The client/server test environment utilizes the same clients and uses a different server depending on the test. The clients are HP Vectra ES/12s with 640Kb memory and 40 Mb hard disks. They each have a ThinLAN (ie-4) card and a StarLAN 10 (PC Link II) card. The StarLAN 10 card is used whenever possible. This test environment is isolated from the other general LANs in use. This ensures that other LAN traffic does not interfere with our testing. Figure 1 shows our test environment.

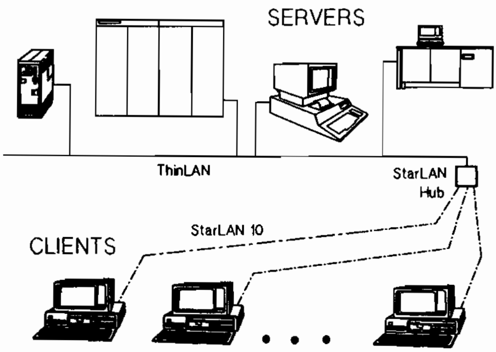


Figure 1

The servers under test are monitored for resource consumption (CPU, disk, memory, etc.) where possible.

The Novell Netware/386 testing was performed on an RS25C with eight megabytes of memory, 103 megabyte disc and a Novell NE2000 networking card. The client software was

Netware 286 version 2.15. The 386 client software was not used due to lack of drivers for HP networking cards.

The HP LAN Manager/X testing was performed on an HP 9000 model 835 with 32 megabytes of memory, one 7963B and two 7935 disk drives, and HP-UX version 7.0. The clients software was OfficeShare version B.00.02.

CONCLUSION

Now that you understand what benchmarks we used and how the tests were executed, you should keep this article close at hand to be referenced when reading other articles by DMSD.

Case Study: Migrating DS to NS, MPE/V to MPE XL and X.25 to ThinLAN

by
Mary Fleischman
Washington Community College Computing Consortium
3860 159th NE, Suite 150
Redmond, WA 98052
(206) 881-4400

1.0 Introduction

The Washington Community College Computing Consortium (WCCCC) was created in the 1970s by joint agreement of the presidents of the 28 community colleges in the State of Washington to provide common computer applications software and computer systems support for all administrative functions of the community colleges of Washington State. In 1983, the community colleges chose Hewlett-Packard as the successful vendor for the implementation of a state-wide, distributed network for administrative computing. As a result, the WCCCC installed 33 HP3000 series 39/40/42/48 processors at 14 locations connected via a centrally managed, private state-wide X.25 network using Dynapac switches and Hewlett-Packard's Distributed Systems (DS) X.25 network operating system. The WCCCC currently administers and manages the largest higher education administrative computing network in the United States.

With a staff of 10-15 programmer/analysts, the WCCCC produced computer applications software for Financial Management (FMS), Student Management (SMS), Payroll/Personnel Management (PPMS), and Production Management (PMS) using a combination of IMAGE, KSAM, VPLUS, TRANSACT (and FASTRAN), PROTOS, COBOL, SPL and Pascal (see Figure 1). Additional software, written in COBOL and SPL, was purchased from Computer Options Company (COCO) for Student Financial Aid.

Figure 1.

Application	Language	# of Programs	Lines of Code
FMS	PROTOS	360	345,000
	TRANSACT	154	..125,000
	COBOL	8	10,000
SMS/FINAID/SMIS	PROTOS	543	520,000
	COBOL	47	62,000
PPMS	PROTOS	383	365,000
PMS	PROTOS	5	5,000
	TRANSACT	7	11,000
	COBOL	4	6,500
	SPL	25	3,500
	Pascal	1	1,200
Utilities	PROTOS	25	25,000
	TRANSACT	18	15,000
	COBOL	40	65,000

A couple of years ago the WCCCC received authorization from the State of Washington to market and sell WCCCC applications software outside the community college system. Through contract with a marketing agency the software was named CEI Plus and a marketing effort was initiated. CEI Plus software is now installed at several locations across the country.

After several years of operation, many of the colleges began experiencing system performance problems that indicated the need for a system upgrade. At about the same time, HP announced the long-awaited Spectrum series of HP3000 processors. With encouragement from HP, we recommended that the colleges consider migrating to HP's new line of processors (rather than upgrading to a Series 70) and the migration process for the community colleges of Washington State was launched.

2.0 Cart Before The Horse

Although HP provided performance metrics for their new line of processors, there was a significant amount of skepticism about the applicability of HP's figures to the processing environment of the community colleges. The only way to guarantee the performance level of the new line of processors, with native mode applications, for the community colleges was to conduct a benchmark of our applications on the Spectrum series. This raised a serious question, "How can the performance of migrated software be measured before it is migrated?" Obviously, it can't be.

So, before we knew it, we were engaged in the process of migrating our applications software to MPE XL native mode. Not having a Spectrum system at our site was the first, and possibly the most difficult, problem we were to face in the entire migration process. Thanks to the local HP office in Bellevue, WA, we were granted the use of a Series 925 and a Series 950 located at their offices, a 30 minute drive from WCCCC, to perform the applications software migration. Using a "granted" system is not recommended, although it's better than no system at all. From day to day we could not be guaranteed that our accounts, and therefore files, would be in existence, nor could we be guaranteed that they had been backed up as one would expect in a software development environment. More than once, we had to re-create source code changes, job stream changes and/or directory structures as a result of "co-existing" with other users of the system.

Aside from the inconvenience of sharing a system, we found the local HP office to be extremely helpful in the migration process, providing us with working space, system resources and migration support in tracking down application specific problems. We couldn't have done it without their involvement. Of course, the promise of selling 25 or more Spectrum processors provided ample incentive for them to help us conduct a successful benchmark.

3.0 Pre-Migration Migration Plan

We devised a plan to perform this migration as follows:

1. Identify the most-used programs from each application area for on-line and batch processing. These would be migrated to native mode.

2. Establish criteria for benchmark success of three second response time for on-line transaction processing and a 100% improvement in batch throughput.

3. Arrange for one of the community colleges to participate in the benchmark, i.e., provide the staff resources for on-line transactions, provide the data bases and data files.

4. Arrange with HP to use their Spectrum processors to migrate the programs, and acquire the appropriate software needed to perform the benchmark.

5. Identify the staff resources at WCCCC who would perform the software migration.

6. Since PROTOS is a COBOL generator, and WCCCC does not save COBOL source for PROTOS programs, regeneration of all COBOL source code for PROTOS programs was necessary.

7. Set up the development and test environment on HP's Spectrum processor.

8. Load our source on to HP's processor and perform the software migration.

9. Test the native mode benchmark against a compatibility mode benchmark for accuracy.

10. Conduct the native mode benchmark and report the results.

11. If the benchmark succeeds, migrate the remaining applications software.

4.0 Application Migration Problem Areas

Three sources of information proved to be very valuable in determining the problems we would encounter in migrating the applications software. One was the Migration Planning Tool (MPT) provided by HP. This tool scans source code, load modules, segmented libraries and job control to identify any intrinsics, PRIV mode calls, commands or other features of MPE that are known to cause problems in migration. The second was the Migration Guide for COBOL which explained the differences between COBOL/V and COBOL/XL run time features. The third was the information we could glean from the Interex conference and the SIG on System Management newsletter. Some of the problems, however, were discovered only by going through the migration process.

4.1 Well-Known Problem Areas

The following problem areas were identified through the use of MPT:

1. PRIV mode calls will not work. Actually, this isn't entirely true as some PRIV mode calls work exactly the same in MPE XL. However, this seriously affected us which I will describe in detail later.

2. CREATE and CREATEPROCESS intrinsics had changed such that some parameters needed to be 32-bit instead of 16-bit parameters. We make extensive use of process handling and many programs required changes for these intrinsics.

3. .LOC. and other language specific calls related to memory address locations (such as BADDRESS and WADDRESS) had been changed to incorporate 32-bit addresses. This sounds simple enough, but the change from 16 to 32 bit words caused many other problems which I will describe later.

4. All SPL source code would have to be converted to a language supported in native mode MPE XL (in order to run in native mode). This raised the question of "C" or Pascal. We decided on Pascal for two reasons, a) we could get Pascal from HP for both our MPE/V and MPE XL systems, and b) Pascal was the language I was most proficient in and, as usual, it needed to be "done yesterday".

5. The FCONTROL intrinsic had changed and certain parameters were no longer supported. This had no impact on us.

4.2 COBOL/V to COBOL/XL

After encountering many problems with COBOL programs aborting with "Illegal ASCII digit" errors (we came to call it the "711 error"), we finally looked more closely at the COBOL Migration Guide. This is a great source of information on COBOL migration problem areas which bears some close reading. From the guide we (finally) learned that COBOL/XL has a completely different way of handling run-time errors than COBOL/V, as follows:

1. If an MPE XL COBOL program encounters a byte that is not immediately translatable into ASCII it will cause the program to fail. With MPE/V COBOL the program would make it's best guess, convert the value to ASCII and continue. This may seem trivial, but if the data is not absolutely "clean", MPE XL report programs will abort for this reason. One solution is to use the COBRUNTIME variable provided in MPE XL to determine how certain run-time errors should be handled.

2. If an unsigned value appears in a signed field (or vice versa), COBOL will determine that this is an ASCII translation error at run-time. This situation could easily be created by entering a value in QUERY as unsigned, but the program has it defined as a signed value. Again, the COBRUNTIME variable will affect the results of this condition.

Another difference to note is that COBOL/XL does not have the BOUNDS or VALIDATE compiler options on by default. To enable BOUNDS and VALIDATE checking in COBOL programs, these options should be set ON when compiling programs (see Figure 2).

4.3 TRANSACT/V to TRANSACT/XL

A major problem area was migrating TRANSACT programs. There were two reasons for this:

1. For the most part, the TRANSACT programs were the oldest programs in the applications software and had been written by programmers not very knowledgeable about TRANSACT, a language quite different from every other language I've worked with. Therefore, the code contained numerous abuses of the language which, mysteriously, didn't bother the TRANSACT/V compiler or MPE/V run-time library, but generated compiler errors in TRANSACT/XL or caused programs to behave quite differently from what was expected.

2. The above condition wouldn't have caused too much trouble by itself, however, we were dealing with a very young and growing compiler with many problems of it's own. It became very difficult to separate the bugs in our code from the bugs in the compiler's code. As a result we worked very closely with the local HP office to get the best version of the compiler as soon as it became available and received direct support from the factory for TRANSACT migration problems.

TRANSACT/XL is now a mature product and is much more robust than TRANSACT/V. The major concern for TRANSACT migration today is that bad TRANSACT code must be cleaned up in order to compile and run programs in native mode. TRANSACT/XL enforces the rules that are stated in the documentation but that are not always enforced in TRANSACT/V.

We also removed all !SEGMENT compiler directives from the source code before compiling in native mode. The documentation states that TRANSACT/XL will ignore the !SEGMENT compiler directives, but we found that not to be the case.

4.4 PRIV Mode Code

The Production Management System (PMS) is the only application that uses PRIV mode and then only for calling the low-level intrinsic DIRECFIND. Unfortunately, the system directory is one of the things that changed in MPE XL. It has been expanded to accommodate 16-character user, file, group and account names (which hasn't actually been implemented yet) and possibly other changes.

Try as we might, we couldn't beg, borrow or steal the information from HP that was needed to implement the DIRECFIND call in MPE XL. HP's policy at the time regarding MPE XL was not to release internals information, no matter what. Their response to us was that we should change the way our software works. In our view, this was not possible. This situation has gladly changed with the release of Architected Interface, which will be available in MPE XL 2.1.

As it happened, another serious problem led to the answer for the DIRECFIND call. It was discovered, soon after we had committed to the benchmark and migration process, that we did not have source code for 20 or so SPL subroutines used in PMS. A well-known consultant, living in the area, was hired to help recreate the SPL source through the use of a decompiler. One of the routines that needed to be recreated was the routine that made the call to DIRECFIND. Fortunately, the consultant was able to get the necessary information through contacts in the 3rd-party software community.

4.5 SPL to Pascal/V and Pascal/XL

Once the missing SPL code had been regenerated, it needed to be rewritten in Pascal. The only real problem in doing this was deciding how to handle arrays defined as both logical and byte arrays in SPL. Pascal's mechanism for defining variable record structures does not lend itself to direct translation and the strong type checking always seemed to get in the way.

Pascal/XL has some MODCAL extensions that make it much easier to work with pointers, but requires a little getting used to. The documentation is not all that helpful either. These extensions aren't very useful for programs that will run in both compatibility mode and native mode since they aren't supported in Pascal/V.

One significant difference between Pascal/V and Pascal/XL is the method of employing error handling. For Pascal/XL you need to $INCLUDE$ a library of error handling routines at compilation time and figure out which routine to call for different types of errors. This is documented in one of the Pascal/XL reference manuals.

4.6 Word Alignment, Data Typing And Subroutines

It was well known that one major difference between MPE/V and MPE XL is the word size, it changed from 16 bits to 32 bits. What wasn't known was just how it would affect the way each compiler handled the alignment of data in memory. Countless hours were spent tracking down weird program behavior only to have this brought home to us again and again. Each compiler works just a little bit differently so it is time well spent to become familiar with how the compiler being used handles data alignment and how alignment will be different in the XL environment.

For example, with COBOL it's best to have all parameters passed to intrinsics or subroutines defined as 01 or 77 level data items so that they are aligned on word boundaries. In Pascal, packing records and arrays no longer guarantees bit alignment if a word boundary is crossed or integer and subrange types are being used. There is an extension in Pascal/XL for specifying CRUNCHED arrays and records which may provide the required alignment.

In some cases, data type changes are required, especially in the case of integer and subrange data types. Some of the intrinsics now require 32-bit integer parameters such as CREATEPROCESS and CREATE. In Pascal/XL, the subrange of -32768..32767 no longer obtains a 16-bit integer. You must use the predefined type of SHORTINT to get a 16-bit integer type in Pascal/XL.

We experienced many problems with programs that call subroutines in a different language, especially TRANSACT calling COBOL subroutines or COBOL calling TRANSACT subroutines. In each case, it was a matter of determining how each compiler was expecting the data to be aligned and making sure they were attempting the same alignment.

4.7 Maintaining Two Source Libraries

In the event the benchmark was successful and the community colleges agreed to migrate to Spectrum processors, we expected the migration process for the entire community college system to be a slow, step-wise process. It was not practical to attempt a wholesale migration for all colleges. As it turned out, the installation schedule ranged from April, 1989, through December, 1990. Also, several CEI Plus client sites were still running MPE/V systems. Therefore, we determined it would be necessary to support the applications software for both MPE/V and MPE XL for the foreseeable future.

In order to accommodate the differences required in the source code for MPE XL we adopted the use of compiler switches, maintaining only one source code file for each application load module or subroutine. We selected "X0" as the switch to indicate native mode. Some examples of how this is used can be seen in Figure 2. To keep things as simple as possible, it was decided to program for MPE/V, making only minor changes as required for native mode execution. Only after all colleges are on Spectrum processors will new features available in MPE XL be incorporated into the applications software.

All program development is done on a Series 950 Spectrum processor. Each source module is first compiled with the native mode compiler and linked into a load module or XL. Then it is compiled with the compatibility mode compiler and prepped into a load module or SL. Both versions are tested on the Series 950. In some cases, it is not possible to test the compatilibity mode load module on the Series 950 because of system dependent code. The compatibility mode load module or SL routine is then moved to a MPE/V development system and, ideally, undergoes further testing prior to implementing on a production processor.

The compatibility mode version does not always get tested on the MPE/V processor, however. We finally got bit on this when a few of our large VPLUS programs blew up with stack overflows after being put into production on MPE/V systems although they had been tested in compatibility mode on the Series 950. As it turns out, the compatibility mode user stack is 400 words larger than the user stack in MPE/V. The solution for this was to decrease the CMSTACK parameter in SYSGEN by 400 words on the Series 950 and reboot the system. And then figure out how to reduce the stack size of the VPLUS programs.

Figure 2.

Example COBOL compile job:

```
!JOB jm6050cp,<<user.account>>;outclass=lp,1;pri=es
!file jm6050l;dev=lp,1
!continue
!purge usjm6050
!continue
!purge omjm6050
!continue
!purge cmjm6050
!continue
!purge lmjm6050
!continue
!cob74xl jm6050c,omjm6050,*jm6050l;&
!  info="$CONTROL VALIDATE,BOUNDS$SET X0=ON"
!showjcw
!if jcw < fatal then
! setjcw jcw = 0
! continue
! link from=omjm6050;to=lmjm6050;rl=rl.nmrl;&
!    cap=ia,ba,ph,ds,mr,pm
!endif
!setjcw jcw = 0
!continue
!cobolii jm6050c,usjm6050,$null;&
!   info="$CONTROL USLINIT$SET X0=OFF"
!if jcw < fatal then
!  setjcw jcw = 0
!  continue
!  prep usjm6050,cmjm6050;rl=rl.rl;&
!    cap=ia,ba,ph,ds,mr,pm;maxdata=31232
!endif
!eoj
```

Example COBOL source code using 32-bit integers for native mode and 16-bit integers for compatibility mode:

```
01  CITEMS.
$IF X0=ON
  07 CITEM1     PIC S9(9) COMP  VALUE 3.
  07 CITEM2     PIC S9(9) COMP  VALUE 9.
  07 CITEME     PIC S9(9) COMP  VALUE 0.
$IF X0=OFF
  07 CITEM1     PIC S9(4) COMP  VALUE 3.
  07 CITEM2     PIC S9(4) COMP  VALUE 9.
  07 CITEME     PIC S9(4) COMP  VALUE 0.
$IF
```

Example Pascal compile job:

```
!JOB zx5022cp,<<user.account>>;outclass=lp,1;pri=es
!file zx5022l;dev=lp,1
!continue
!purge uszx5022
!continue
!pasxl zx5022,omzx5022,*zx5022l;info="set 'x0 = true"
!showjcw
!if jcw < fatal then
!  setjcw jcw=0
!  continue
!  linkedit
rl rl.nmrl
purgerl module=zx5022
addrl from=omzx5022
exit
!endif
!setjcw jcw = 0
!continue
!pascal zx5022,uszx5022,$null;info="set 'x0 = false"
!showjcw
!eoj
```

Example Pascal source code using different compiler options and creating SHORTINT type for Pascal/V:

```
$subprogram$
$if 'not x0'$
$segment'zx5022',uslinit$
$endif$

program zx5022_prog;

 type

$if 'not x0'$
  shortint    = -32768..32767;
$endif$
```

4.8 3rd Party Software

The WCCCC employs several 3rd party software products in the development and support of its applications in the community college system. These include: ESPUL from RAC, MPEX from VESOFT, DBGENRL from Bradmark, PROTOS from PROTOS, DataExpress and LYNX from IMACS, FASTRAN from Performance Software, S/COMPARE from Aldon and QEDIT from Robelle.

Most of the applications software developed by WCCCC is written in PROTOS, a high-level COBOL generator. We contacted PROTOS about our migration plans and acquired a version of PROTOS that would run in compatibility mode on Spectrum processors. Several problems with the run-time library were encountered during the migration process which required getting new SL routines shipped from

PROTOS. We are now using a native mode version of PROTOS and all problems related to PROTOS migration have been resolved.

Another software product important to our applications software is DataExpress from IMACS Corporation. IMACS supplied us with a compatibility mode version of DataExpress which runs on Spectrum and MPE/V processors. We have been beta testing native mode DataExpress and hope to have it installed on our Spectrum processors soon.

No other problems have been encountered with 3rd party software products. However, it is important to contact 3rd party software vendors as soon as possible in order to minimize problems during migration.

4.9 PTOP to NetIPC

Program-To-Program (PTOP) communication was a facility within HP's Distributed Systems Network (DS) that allowed programs to communicate with each other over the DS network. HP informed us that they did not migrate PTOP to the Spectrum environment. That meant that PTOP applications would not work in the Spectrum environment, period. We heard rumors that PTOP was actually there but not supported. Of course we tried it, and not enough of it was there to support our application. It just didn't work.

Thus began a crash course in Network Inter-Process Communication (NetIPC) and Remote Process Management (RPM). I discovered a very instructive appendix in the NetIPC 3000/XL Programmer's Reference Manual P/N 5958-8600 entitled "*Migration from PTOP to NetIPC and RPM*". The examples were in Pascal which made it even easier to try them out. Unfortunately, the examples have some problems, but with a little common sense and experimentation they can be solved. The next challenge was to convert the Pascal examples to COBOL for incorporation into our applications software. It wasn't that difficult, but more cumbersome to implement in COBOL than in Pascal.

A few lessons learned: a) explicitly create a remote session with a REMOTE HELLO command before trying to establish RPM, it seems to work better that way even though the documentation states it isn't necessary; b) keep regular messages going back and forth between the client and server (master and slave) because disabling the socket time-out doesn't seem to work in all cases; c) in situations where there will be no messaging activity for a significant period of time, i.e. more than a minute or two, shutdown the socket and re-open it when it is needed again.

NetIPC/RPM requires significantly more code, and more complex code to implement than PTOP, just to imitate the PTOP function. NetIPC/RPM, however, is much more powerful than PTOP and has capabilities that could provide much more sophisticated networking applications.

5.0 System Migration Problem Areas

Ok, so you've got your application migration all figured out and you're ready to install a system. But wait a minute, what about all of those utilities you're used to, are they going to work the same? Will you need to change your network? Will your operator know what to do?

With MPE XL up and running there seems to be very little difference between MPE/V and MPE XL. The differences are there but you have to look to find them in most cases. The System Manager and System Operator have a very different system to work with and the transition from MPE/V to MPE XL requires some planning, training and procedural changes.

One of the best methods for preparing for the operational changes involved in moving from MPE/V to MPE XL is to arrange to have a Spectrum system installed on site a month or more prior to the first production system installation. It is very instructive to test out the operational environment, to determine the procedures and utilities that will need to change, and to prepare for any training you will need to acquire or deliver to operations staff.

5.1 System Operation

The operations staff was sent to HP System Operator training classes to learn the new commands for controlling the system. The changes translate as shown in Figure 3.

Figure 3.

MPE/V	MPE XL
CTRL A> =SHUTDOWN	CTRL A> =SHUTDOWN The messages are very different from what you would see with MPE/V. After the last message is displayed, then the system must be halted as with the command shown below
CTRL B>HALT	CTRL B>TC or RS TC stands for Transfer Control and will perform a soft reset of the system. This command is generally preferred as it takes less time and preserves the system for a memory dump if necessary RS stands for Reset and will perform a hard reset of the system and run a memory self test
LOAD	At CM prompt type TC or RS and boot from alternate path (TAPE)
RELOAD	INSTALL - replaces all information on disk with what is on the boot tape (System Load Tape or SLT) and is followed by a full restore of files
LOAD	At CM prompt type TC or RS and boot from alternate path (TAPE)
UPDATE	UPDATE - replaces the operating system with what is on the tape and preserves the CONFIG group files which include the I/O configuration.
LOAD COLDLOAD	{No equivalent procedure}

START	At CM prompt type TC or RS and boot from primary path (DISC)
COOLSTART	START NORECOVERY This uses the configuration stored in the group CONFIG.SYS to create a system configuration and loads MPE XL from disc
START	At CM prompt type TC or RS and boot from primary path (DISC)
WARMSTART	START RECOVERY This uses the configuration already loaded into memory, recovers spoolfiles and loads MPE XL from disc

The other system changes that were significant included:

1. The serial connections (terminals) are not provided via system ports but via a Distributed Terminal Controller (DTC) which is attached via coax cable to a LANIC card installed in the processor. The system has Network Services and ThinLAN Link software installed to service the DTC on the LAN. For a system-to-system LAN, a separate LANIC card is required. The DTC LAN and the System LAN can share the same coax backbone provided the Spectrum system is running MPE XL 2.0 or later. In order to reset ports, use the command SYSDIAG and then RUN TERMDSM. After that provide the DTC number, the SIC (Serial Interface Card) number, the Port number or the Logical Device number to perform the reset.

2. To take a memory dump the procedure is as follows:

   CTRL B>TC
   {Other questions until you got the ISL prompt and entered}
   ISL>DUMP

3. Doing a backup consists of performing a STORE, not a SYSDUMP. The DIRECTORY option should be included on the STORE command to copy the system directory to tape along with the files. Since STORE does not include a copy of MPE XL or the system configuration files, it is also recommended that a System Load Tape (SLT) is created at backup time.

5.2 Distributed Systems X.25 (DS) to Networked Systems X.25 (NS)

When the migration process began, the WCCCC state-wide network consisted of 25 or more HP3000 systems running DS software over private X.25 network. Spectrum systems do not support DS software. This meant a major change for the WCCCC network.

Conversion from DS to NS is a major task for a network this size. Originally, we estimated 400 staff hours would be required to complete the migration from DS to NS on MPE/V. For each MPE/V system on the X.25 network the following steps were needed:

1. Install NS/3000 and NS/X.25 software SUBSYS tapes with appropriate patches. This required a minimum of four hours of downtime for each processor to complete the software installation.

2 Acquire a C class IP (Internet Protocol) address set for each subnet in the network. HP provides these. In our case, we needed one set for the X.25 network, one set for the LAN at WCCCC and one set for each Spectrum system at a remote site (I will explain why later).

3. Set up the NS configuration files on each processor in conformance with DS using NMMGR. This process requires about 1 hour for each processor (once you've got it right).

4. We had to use the shared INP method (NS and DS on the same INP) because we would be migrating one system at a time necessitating combined DS and NS activity. For this, a system reconfiguration is required, about one hour of downtime for each system.

5. When everything is in place, the DS/NS software switch must be enabled in both the DS and NS configuration, the system shut down and coolstarted.

That takes care of X.25 for MPE/V. In order to use our resources most wisely, we decided each remote system would stay on DS until their Spectrum processor was installed. Some of the processors located at WCCCC, those that had the longest wait for migration to Spectrum, were migrated to NS X.25.

5.3 Adding a LAN and Network Services/XL

X.25 link was not available for Spectrum processors at the time we began production processor migrations. Therefore, it was decided to put all Spectrum processors on a IEEE 802.3 LAN at WCCCC as they were installed and use a Series 52 as a gateway between the LAN and the X.25 network.

Installing NS on a LAN is fairly straightforward. Problems arose in the installation of the gateway between the LAN and X.25. These problems mostly stemmed from incorrect information from HP on two points: 1) assigning IP addresses and 2) using network directories. We started out with a four-system network, two MPE/V systems on X.25 and LAN (one of them was the gateway), one Spectrum system on LAN only and one MPE/V system on X.25 only. We were using one B class IP address set for the entire network, subnetting on the B class IP addresses, and using the gateway system as a probe proxy for each subnet. When we started the network, we could only get one subnet at a time to work.

It took the network escalation team from the local HP office many days to solve the puzzle. They finally learned from the factory that NS did not support subnetting on B class addresses, nor would probe proxy work for subnets. Each subnet required a unique IP address set and each node in the network needed to have a network directory of all nodes and IP addresses in the network. These simple changes made, the network came up without a hitch.

For Spectrum processors at remote sites, HP provided a product called the XL Server, which consisted of a very low-end HP3000 MPE/V system pre-configured to serve as a gateway between the X.25 network and the Spectrum processor. The XL Server and Spectrum system were on an IEEE 802.3 LAN that is treated as a subnet by NS, and therefore required a unique IP address set. At one point there were as many as 10 gateways configured in the network, all but one of which served a single Spectrum processor on a LAN.

5.4 DS/NS Incompatibility

We had essentially created a dual network environment. The NS network was separate from the DS network but "joined" by the MPE/V system which had the DS/NS switch (Dual INP) configuration.

From a Spectrum system you could gain access to a MPE/V system on DS through the network by first doing a remote logon to one of the DS/NS systems.

One unfortunate fact was discovered. It is not possible to run block mode screen programs using Virtual Terminal Service going from a NS only system to a DS only system. To run block mode programs using Virtual Terminal Service on a DS only system, the home system must be running DS. This posed a serious problem since it was necessary for staff to have this capability to support the applications software on all systems in the network.

HP had no plans to correct the incompatibility between DS and NS since DS is a dead product. The solution, presented by HP, was to migrate all of our systems to NS. Suddenly we were faced with the 400+ hour migration project needing to be completed "yesterday". Needless to say, we were not very happy with HP's response to this problem. It was finally resolved to keep one of the MPE/V development systems, a Micro/XE to minimize hardware support costs, and give each member of the staff who required it two terminal ports, one for the Spectrum development system and one for the MPE/V system.

5.5 Multiplexors, Terminals, PCs and Serial Printers

For the systems located at WCCCC, CODEX multiplexors and modems are used to provide data communications to the college campuses. Most of the terminals are HP2392A terminals and the PCs primarily use Reflections software from Walker, Richer and Quinn. There have been relatively few problems related to the use of this equipment on Spectrum processors.

A couple of things to note are: 1) Terminals and terminal emulation software need to be configured for XON/XOFF Transmit and Receive Pacing with 8 data bits and no parity, 2) Reflection software should be updated to version 3.2 or later, 3) Flow control may need to be disabled on multiplexors, although this will depend on the brand of multiplexor. We found that flow control on the multiplexors interfered with block mode applications.

Serial printers should also be configured as 8 data bits and no parity. The only real problem we had with serial printers was a problem with VFC download files. This problem occurred in MPE XL 1.2 and was corrected in MPE XL 2.05.

5.6 Not All Utilities Migrate

Every operation is different, but it is fairly common to find a fair number of contributed utilities among the tools used by System Managers and Operators to operate, manage and use HP 3000 systems. Unfortunately, many of the most useful utilities used in our operation have not been migrated to MPE XL. That's why having the Spectrum system ahead of time was so useful. We re-discovered all of the utilities we used and took for granted. In most cases, it was just annoying to have to learn to live without them.

There was one utility that we really couldn't operate without and that was the ALLOWME utility from the Boeing Tech Account. Many of the community college systems that are located remotely are operated by staff at WCCCC after normal business hours. The operators need access to console commands and MPE will not allow the console to be switched to a virtual terminal (with good reason). We put pressure on HP to help us find a solution to the problem. Eventually, we learned what Boeing was using as a substitute for ALLOWME.

The following command is added to the SYSSTART file on the system:

ALLOW @.@;COMMANDS=ALLOW,DISALLOW

In addition, a system logon UDC is installed which executes a command file that ALLOWS console commands based on the logon id of the session or job logging on. The final step of the command file is to DISALLOW the ALLOW command. This is actually better than ALLOWME because you can tailor the set of console commands allowed to a given signon instead of giving them all to anyone with OP capability.

6.0 Installing The Production System

Successful installation of a production processor with minimum downtime became our next project. An installation review team was identified and an installation checklist was developed for each installation. The plan included all steps that must be completed during the installation process, ranging from site prep to testing of the applications prior to allowing users to logon to the system.

The installation review team met several weeks before the installation to review all details and resolve any questions or conflicts. Then about two weeks before the installation the review team would meet with a representative from the college to review with them the steps that would be required for the installation. Each college was requested to reserve three working days for the installation although, in nearly all cases, it was completed in less than two days.

A Systems Support representative of WCCCC would be on-site while the HP Customer Engineer installed the hardware and loaded the operating system and subsystem products, verifying the system configuration and proper installation of the software. This proved to be very important to ensure that the system was installed properly and all components were functional prior to letting the CE leave. The Network Administrator from WCCCC would configure the network and verify its functionality. Then the Systems Support representative would create the accounting structure, restore all the files, install the native mode library and any utilities and prepare the system for operation.

7.0 Conclusion

The WCCCC is still involved in the migration process and facing new problems everyday related to the changes required in moving from MPE/V to MPE XL. The last Spectrum system installation is scheduled for December, 1990. Although we have invested a significant number of staff hours in this project, we feel we have gained a great deal from our investment and consider this project to be highly successful.

The user community is very pleased with the performance and reliability of the new systems and we have experienced very little down time related to system problems. The majority of ongoing problems are related to serial communications through the DTCs, managing a large NS network and coping with the changing environment as HP introduces new products and new ways of managing and operating systems and networks.

The WCCCC has grown tremendously through this experience in terms of increased processing power and the options available to us and the users in making new and better use of the computer systems and applications. In the final analysis, we are very satisfied with the decision to migrate and feel this puts the Washington Community Colleges in a very good position for meeting the increasing computing and networking needs of the next five to seven years.

James Cohen
Data Processing Manager
Mecca Leisure Ltd
Mecca House
76 Southwark Street
London SE1 0PP

Telephone :- (44) 71 928 2323

MIGRATION FROM THE DATA TERMINAL CONTROLLER

With the introduction of Precision Architecture came the arrival of the DTC, then called the Data Terminal Controller. This replaced the ATP and ADDC used on the Classic Systems.

The strategic change was that the terminal controller was no longer connected directly to the processor, but connected to either a Thick or Thin LAN.

The DTC itself was configured using the NMMGR program. Beyond the fact that the DTC's could be located outside of the computer room environment (by extending the LAN), the DTC did little if anything to improve the life of the Computer Manager. In fact, because of bugs, port locks and lack of power fail recovery routines, the first release of the DTC was anything other than a delight to manage. In fact on first release the only saving grace of the DTC was that it enable a greatly simplified wiring strategy, if you were in the privileged position of being able to totally rewire your building as part of the migration. Since you could now use horizontal twisted pair wires terminated at patch panels for either terminal/DTC access or personnel computer/StarLan networking.

For those of us already dependent on X.25 the arrival of the HP3000 Series 900 systems, was a problem. The INP on which we had all come to rely on was no longer available. Soon after the HP3000 Series 950 started to ship in any quantity Hewlett Packard offered the X.25 Server - which was little more than an HP3000LX, with an INP installed, for those who required HP3000 to HP3000 communications over X.25. For those of us with large terminal populations, connected either via private PAD's (such as the HP2334/5) or via the public data networks (such as Telenet, PSS or Transpac), we could easily make use of X.25 by placing PAD's in front of DTC's and configuring the ports as US Modem ports.

However, both these solutions were of little functionality, but since Mecca Leisure was interested in connecting a terminal population of two hundred remote sites to an HP3000 Series 950, this paper will restrict itself to the problems involved in implementing DTC 2.0 in this type of scenario.

So let us start by looking at the situation at the start of the migration. Mecca Leisure had an HP3000 Series 950, an HP3000 Series 925LX and an HP3000 Series 70 connected on a single LAN. Connected to the HP3000 Series 925LX was a single DTC and an HP2335 - this was our development system and in effect mirrored the activity on the HP3000 Series 950.

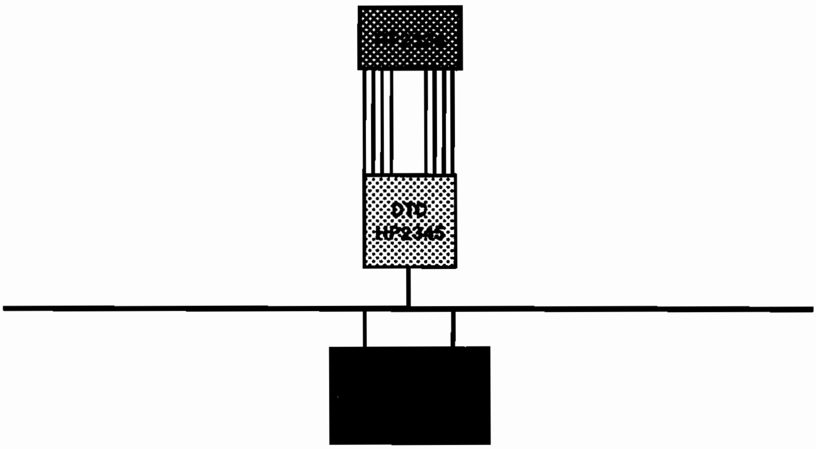


Fig 1

The main production system was the HP3000 Series 950 with five DTC's, four HP2334's and 32 X.25 ports. Since the fastest line the HP2334 would support was the 9600baud X.25 circuit, it was felt that the maximum each line could support was 8 terminal users (in fact the maximum loading on these lines never exceeded 35%). Also connected was a large terminal population that required access to all three systems.

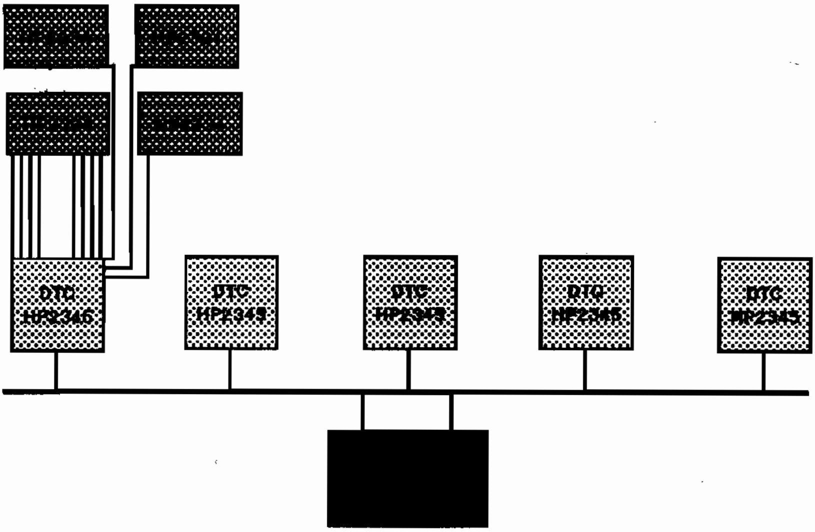


Fig 2

And finally the HP3000 Series 70, that was being used for Electronic Mail.

The original network had been designed around the requirement that 80 branches would require access to the HP3000 systems for a maximum of one hour each per day. But less that one year after installing the original network the requirements had changed considerably. Following a large acquisition, the requirement was now for over 200 branches, and each branch would require online access for about two hours per day. Electronic Mail was to be extended beyond Head Office to all the branches (where AdvanceMail was to be installed on each of the Vectra's) and data was to be collected and delivered to each branch each night (requiring a connection for about half an hour per branch).

It was clear that the current network, while it could be extended to support our new requirements, was not a cost effective option, since in the United Kingdom 9600 baud X.25 circuits are relatively expensive when compared to the band width of 48 kbit circuits. Furthermore, we were beginning to become aware of the load caused on the HP3000 Series 950 by users logging onto this system, only to gain access to another system. This situation could only get worse with more branches (and central users) being brought online and with the future introduction of company wide electronic mail - that was to be implemented on an HP3000 Series 935.

The solution appeared to be two fold :-

- Implement data switching in front of the processors
- Install 48kbit X.25 to remove the need for racks of DTC and HP2334/5

Fortunately this was all being planned, shortly ahead of the introduction of DTC 2.0, and our Sales Engineer and Account Engineer were able to trawl enough information from the Hewlett Packard Division to enable us to plan on its introduction, before product announcement at last year's Interex Conference.

Now before looking at the steps needed to move from the original DTC (DTC 1.0) to the new DTC (DTC 2.0), it is best to define exactly what DTC 2.0 is and what functionality it offers you.

DTC 1.0 was called the Data Terminal Controller, while DTC 2.0 is called the Datacommunication and Terminal Controller - and in a nutshell this is the difference. DTC 1.0 simply replaced the ATP, DTC 2.0 provides a complete data communications sub system. It provides data switching capabilities between multiple HP3000 Series 900 systems and advanced networking capabilities in the X.25 arena. To enable one to make use of these functions, one has to purchase a modified ES/12 PC that is configured to run OpenView software, that configures and controls DTC 2.0.

Since DTC 2.0 can act as a data switch between multiple HP3000's clearly it is not practical for the HP3000 to be involved in the majority of the configuration aspects of the device. Therefore as the DTC's no longer "belong" to an HP3000 another device is placed directly on the LAN to control the DTC's and other communication devices.

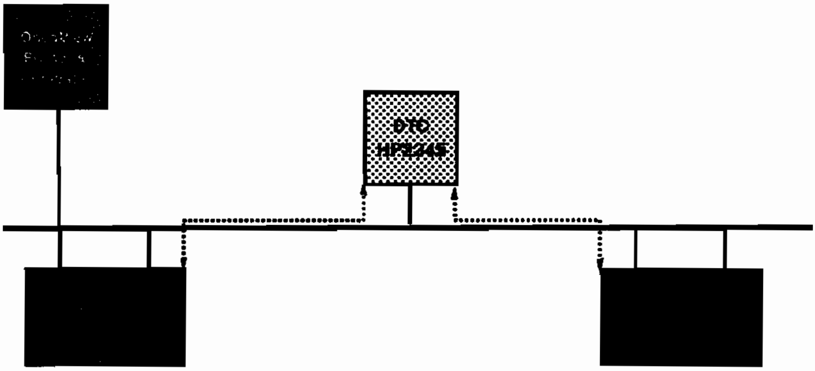


Fig 3

A DTC having six interface slots can support either three pin RS232 ports, five pin RS422 ports, twenty five pin RS232 or a maximum of three X.25 interface cards (either RS232 option, with speeds up to 19.2 kbps or V.25 option, with speeds up to 64kbps); slot zero on a DTC must always contain an asynchronous interface card. The X.25 cards can be configured for either terminal pad or "system to system" communications, although for system to system communications additional NS software is required.

The DTC's can be located anywhere on the LAN and either ThickLAN or ThinLAN, the decision is yours but is really dependent on the environment your LAN is laid in and the maximum length you require, since ThickLAN can be up to 1,500 metres long , while ThinLAN is limited to 555 metres. In both cases two repeaters are required.

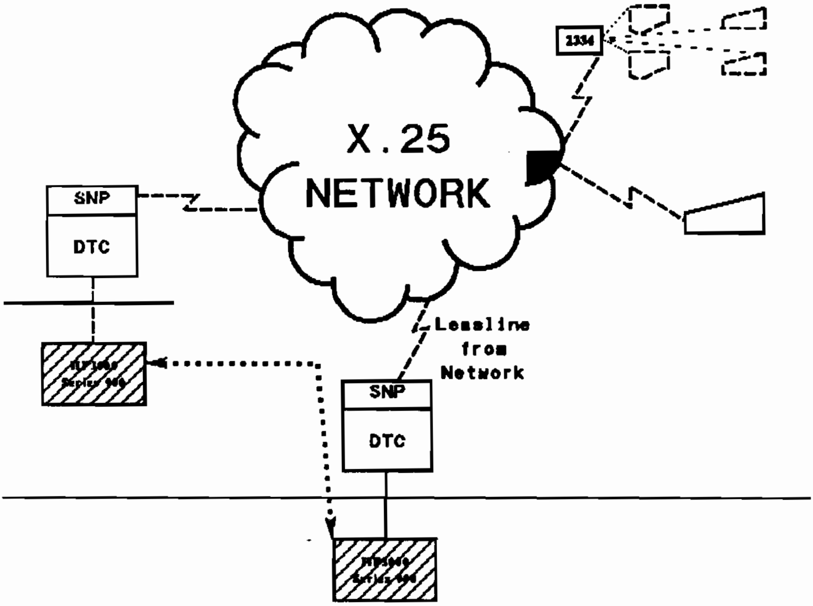


Fig 4

Now that we have an understanding of the networking philosophy, what steps are required to move from DTC 1.0 to DTC 2.0 ?

Well firstly, if you have anything other than a directly connected terminal population, with no software that opens any particular logical device, then I believe this migration needs as much thought as the effort involved in your initial migration to MPE/XL. But having said that the implementation is not as complex.

Since devices can switch between multiple HP3000's they can either be nailed, that is to say that a port can be allocated a specific logical device number (on each HP3000, this is where confusion can commence), or unnailed, that is to say that the system will allocated a logical device number from a pool.

The only devices that must be nailed are printers, they must be allocated a specific logical device number, since without it how can the system spooler work ? However, in theory a printer could be spooled to more than one HP3000. This can happen since the same physical port can be nailed to one or more logical device numbers. What happens is that the first system to FOPEN the logical device will be successful, and should another system attempt to open the device will receive a device not available message. I don't believe that I can recommend you to spool one physical printer onto more than one system - your operators might get a little upset with the number of spooler messages that they would receive. You would be better advised to install a spooler management system (such as Unispool) to move the spool files to the single system on which the printer is configured.

So that is the type of issues that have to be resolved for printers, and issues that are simple to resolve, since a printer has to be allocated a logical device number. But what about terminals ?

The issues with directly connected terminals are, should the device be nailed and should the user have access to the DTC command system ?

The issues for migrating a terminal X.25 population are much the same. However, the only reason to nail the terminals (since you cannot be sure which LDev will be allocated), would be if you require to FOPEN a device. This would be the case if you were polling round your remote terminals, using products such as Horizon/3000 and collecting batch data.

So having considered the issues, what are the immediate differences ?

- For those terminals who are allowed to switch between processors (since there is little point in making switching available to users who only need to access one system), you need to train them on the functionality of the data switch. How to logon and logoff, how to return to the data switch.

- There is an Hewlett Packard "Quick Reference Guide for Terminal Users" (Part Number D2355-90002), you get one with OpenView. It is a piece of A4 card, folded in half, and would be very useful to issue to users who are going to make full use of DTC2.0, but since the list price is £6.50 each - approximately $10 - I would recommend you produce your own.

- Do not expect users to have the same LDev each time they logon, unless every port is nailed. There is a choice to be made here, either nail every port - in which case configuration takes some time, but systems documentation is simple, or leave ports unnailed - but have the problem of systems documentation and complex support.

  Of course, the key criteria in whether to nail or not, is whether you want to FOPEN the port, if you do it must be nailed.

- From an X.25 viewpoint, you can consider the SNP on MPE/XL systems to be identical as the INP on an MPE/V system. Indeed software designed specifically for INP's worked with no migration effort when run on an SNP.

- Maintenance of the OpenView PC is an issue. The PC is in effect an integral part of your HP3000. If there was a total power failure, and your OpenView PC failed to come back up, your system would in effect be down. However Hewlett Packard's highest level of hardware support for PC's (at least in the UK) is four hour response Monday to Friday. Since many of us are required to keep our systems available 24 hours a day, seven days a week, this is not satisfactory and needs resolving sooner, rather than later.

- An additional remote support modem is required for the OpenView PC, it must be fully 100% Hayes compatible (so do not buy an HP internal PC modem). Unlike support modems for the HP3000, you are expected to purchase one, even though without it HP will have difficulty supporting you, this is particularly the case if you have a NetAssure contract - HP would have little alternative but to send an engineer on site.

- The OpenView PC controls your data network. However the configuration details that used to reside only on the HP3000 (within the file NMCONFIG) now also reside on the PC. Therefore it is essential that the PC is fully backed up, and the details stored off site. To that end we have installed a PC internal (Irwin) tape drive, and the OpenView PC backed up to tape weekly.

So now that we have considered the issues and differences, what steps do you need to take to migrate from Host to PC control, with minimum impact on you installed user base ? These are the steps we took, but should not be considered exhaustive :-

- Take a copy of the NMCONFIG file and make all the necessary changes. This can take some time, and does not effect the system configuration, until it is liked with SYSGEN.

- You can now configure the DTC's for OpenView, decide which ports are to be nailed. Also configure the SNP's and allocate LDev numbers for any X.25 port that is to be opened by the HP3000, either to dial out or for a printer connected over the X.25 network.

- Once you have completed the NMMGR configuration, take a print of the configuration and start on the OpenView configuration. This takes some time to complete, since Windows is relatively slow, once you have mastered OpenView - which is very easy to learn and use.

- Start the configuration by drawing the network with OVDraw, and then start the configuration by configuring each CPU board on each DTC. Unfortunately you cannot copy from one DTC to another, and this means that the "Welcome Message" to each DTC has to be input each time.

- Once the CPU are configured, you have to configure each board in each DTC. This is not as difficult as it seems, since once you have configured a port on a board, it is simple to copy and paste this configuration on the remaining ports, and then copy and paste the board to other slots on the DTC.

  Particular care is needed to configure ports that have been nailed on the HP3000, since the configurations must match for the port to work.

- It is now while you are configuring that you have decide, whether or not to enable data switching. Many users only require access to one application or other, and I do not believe it desirable to let them data switch. Therefore, initially we limited data switching to a minimum of users and gradually introduced to more, as training time permitted. This means that for the majority of the users they are not even aware of the networking changes.

- If you are moving users from a 2334 terminal to an SNP, you may well have to issue new communications command files, to reflect the change in X.25 address. If this is the case this can be completed after the migration to DTC2.0. However, one of the difference we found in DTC2.0 over DTC1.0 was that when connected to DTC1.0, once the BYE command had been issued the network was cleared down and the call disconnected. This was not the case with DTC2.0, and unless the users have access to data switching (which is unlikely, since you have not issued new command files), they cannot clear the call, except with an X.25 command "CTRL P CLR". Unless this change is made, the result is lots of users unnecessarily connected to the HP3000 incurring connect costs.

- The X.25 configuration for the SNP, now needs to be completed. This is not complex, but again takes some time. This is particularly the case if you wish to dial out over X.25 since each X.25 address (in our case nearly 200) needs to be defined with its dummy LDev and X.25 address.

- Now that you have made the necessary configuration changes on the P3000 and constructed the OpenView configuration, you can choose your time to go live. Link your configuration into SYSGEN, shutdown the system, power cycle the DTC's, bring up the system. You should now be configured with OpenView.

What has not been covered above is the issue of training yourself, as Systems Manager, and your Operators. At the time of writing Hewlett Packard has not yet offered an OpenView training course, although it is believed that a four day course is being prepared. The documentation supplied with OpenView is simple to follow, as so long as you have had prior experience of MSWindows and configuration of an MPE/XL system, I think it unlikely that you would need to attend any course, and could learn the system within a day of receipt. If you have had no experience of MPE/XL, then you would be attending the Systems Manager's course. This does not cover OpenView which is an additional course.

In conclusion, the migration to OpenView is a relatively simple process, so long as you have considered the impact on your users. If you have a development system, install OpenView there first - there are differences particularly for communications software.

The benefits are clear :-

- Data Switching, that reduces the load on multiple HP3000's.
- Improved X.25 networking
- Simple configuration
- Improved diagnostics

If you have more than one MPE/XL system networked together, OpenView is a cost effective simple tool for the Systems Manager to improve the operation of the computer network.

TITLE: Networking in a Multi-Platform Environment

AUTHOR: Mike Eckstein

The Gap, Inc.

3434 Mineola Pike

Erlanger, KY 41018

606-283-1100

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 7089

NEVER RUN ANOTHER CABLE AGAIN

J. W. Swearingen
The Apex Group, Inc.
7151 Columbia Gateway Drive
Suite F
Columbia, Maryland 21044

(301) 290-1606

Paper Number 7090

In the times of ancient Greek mythology, it is said that a demi-god named Prometheus existed. He was sometimes credited for creating humankind out of earth and water. He was also one to the famed Titans and brother of the "world-on-his-shoulders" Atlas.

Prometheus, as legend has it, stole fire from the gods for the benefit of mankind. Zeus, furious with this stunt, punished Prometheus by having him bound to the side of a great stone mountain where his liver would be devoured by an eagle. Each night his liver would regenerate only to be devoured again and again for eternity.

This myth, while certainly more dramatic, has many uncomfortable parallels with the business of supplying voice and data distribution for our user communities. Cabling, it seems, is an endless series of new cable runs, moves, and changes. Just when we think that we are on top of these changes, a office move is proclaimed or a reorganization takes place and we are thrust into a mad cable scramble trying to respond to these changes. Our documentation during this scramble becomes outdated leaving us vulnerable and confused. And like Prometheus, we are seemingly, condemned to repeat our "fate" over and over again.

Premises Distribution Systems (PDS) are complete interconnect wiring systems that link voice and data communication equipment together into one network. Utilizing a simple modular plug and jack approach, the modular network applies "twisted pair" technology to connect telephones, data communication workstations (such as PCs, synchronous terminals, etc.), switching equipment (PABX's, key systems, etc.), and information processing systems (computers, word processors, etc.) to one another, as well as to other outside networks.

PDS is totally integrated. Each component is designed to work with all others in the product line. All of the cabling and associated hardware required to interconnect and transport voice and data information are combined into one standardized system. The modular design lets all connections, additions, and relocations be made without the use of tools.

During this discussion, I will be attempting to convince you that there is a better way to handle your voice and data distribution needs. This methods is very modular, scaleable, flexible, and upgradeable. Moreover, it will handle just about any kind of voice or data processing hardware available. It will also increase your ability to trouble shoot and monitor your cable plant, and perhaps most importantly, it will possibly prevent your organization from ever having to run another cable again.

Evolution of Voice and Data Interconnection

In 1983 a court ruling put an end to the national Bell Telephone system as we knew it. Until that time, system wiring-design, hardware, installation and maintenance were all a part of the services provided by the phone company. Everything necessary for voice communication throughout the network was provided by Bell. The wiring was owned and maintained by Bell and was "leased" monthly to the individual user. Any work on the system had to be completed by Bell Service Representatives and was included as a part of the monthly phone bill.

During this same time, most organizations had diverse information equipment that required inflexible, expensive and mutually exclusive interconnect systems planned and wired separately from the voice wiring. Consequently, the wiring for voice differed significantly from that data in types of cable and support hardware. As new systems were added, there was no assurance that they would work in harmony with the rest of the network, let alone with the existing telephone lines.

The divestiture of the Bell System further confused an already confusing situation. The telephone company was no longer required to provide and maintain the wiring (for voice) as part of the system. Instead, the entire system now had to be designed and maintained by the individual owners. Once taken for granted, on-premise wiring systems suddenly became the focal point of the wiring network.

Together At Last

For the first time, the connective wiring between voice and data equipment was evaluated as a whole. The advantages of total system integration became obvious. Through standardization, an integrated system meant both present and future connectivity needs could be met, even among different networks. Total integration also meant significant cost savings. It was quickly realized that the less expensive "twisted pair" wiring, typically used for voice only, was also capable of handling the needs of data transmission. This eliminated, in most cases, the need for bulky, expensive coaxial cabling throughout a voice and data network. Also, the modular technology typically used for telephones could now be used for connecting all elements of a network. As a result, twisted pair wiring became the accepted standard in voice and data network interconnection.

Twisted Pair Cabling Displaces Coaxial Cable

Coaxial cable was once considered to be the only way that a voice and data network could be connected successfully, and twisted pair wiring was used exclusively for voice equipment interconnection. After the Bell divestiture, much more attention was given to the wiring system. It was quickly realized that the same twisted pair wiring used for telephone applications could also be used for computer applications. This was considered to be an important discovery because of the many advantages that twisted pair wiring has over coaxial cable:

Less Expensive - Twisted pair wiring is much less expensive by the foot than the heavier, more complex coaxial cable. For instance, one foot of grade 62A/U coaxial cabling runs about $.25 per foot. Compare that with a comparable grade <u>single</u> twisted pair that will cost about $.03 per foot. That's an overall savings of over 90%.

Ease of Installation - Twisted pair uses modular plug and jack connections, identical to those used on today's telephones. This method is the simplest to use. All connections, additions, or relocations can be made without the use of any special tools. More importantly, data changes can be made as easy to as moving a modular telephone. The expertise required to make similar moves using coaxial cable generally is significantly greater, is more expensive, and require more time.

Flexibility - Experts approximate that 50-75% of all workstations (voice or data) are relocated within their first year after installation. This series of moves, changes, and adds is referred to a the churn rate. If the interconnect system isn't flexible enough to meet these requirements, every move will cost. Because twisted pair uses the modular approach, all system moves can be made easily in a matter of hours, not days.

Coaxial cabling, on the other hand, is cumbersome and much more permanent. Connections are made with complex unions and require the assistance of technicians. As a result, changes to the system may take days, or even weeks, assuming an appointment with a technical representative can be arranged.

Appearance - Today's office environment is becoming even more advanced with the continual addition of voice and data equipment. Because twisted pair wiring is lightweight and flexible, each workstation doesn't have to be a complex web of wiring. Instead, the connection can be made discreetly, and is easily hidden from view. Additionally, wallplates supporting voice circuitry are in place already. This facility can be added on to very easily with significant expense or modifications to office walls. This same type of aesthetic value is not available with coaxial cabling.

PDS Components

The Premises Distribution System has a common architecture consisting of a uniform set of components. These include:

Workstation

This subsystem consists of information processing and handling equipment such as a phone and a data terminal. PDS allows for any type of computer hardware or phone system available today in the industry.

Modular Interface Cords

All workstation equipment is connected through a standard modular interface cord. Typically RJ-45 type connectors are fitted with equipment adapter plugs to accommodate individual equipment. For equipment needing coaxial signal characteristics, "baluns" are used for converting unbalanced signals into balanced signals thus allowing operation over common unshielded twisted pair cable.

Premises fiber optic cable is also a very important part of PDS. While industry experts agree that fiber to the workstation will not be practical for several years, many customers choose to install the fiber today to be positioned for future technologies like FDDI.

Premises Distribution Systems from The Apex Group, Inc.

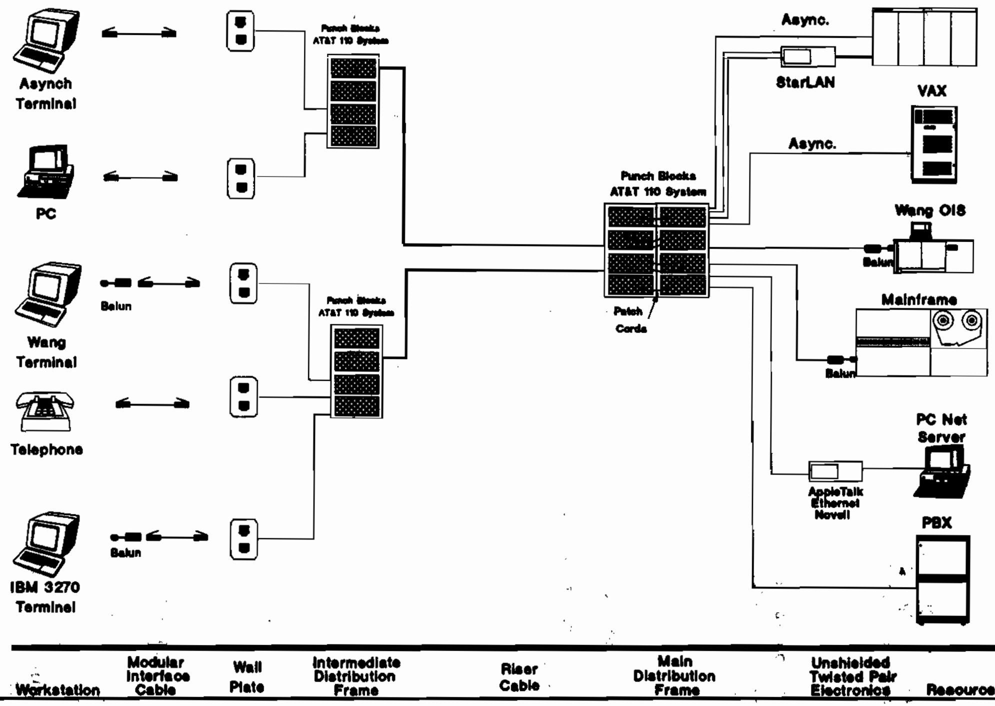


Figure 1 Page 5

Wall Plate

Each workstation location is fitted with a dual RJ-45 type modular wall plate. This allows at least one phone and one data terminal or PC. A full eight pair are run to these wall plates for maximum flexibility and future ISDN compatibility. By using special purpose splitters or quad wall plates, users could easily expand to two phones and two data devices (printer & PC, 2 PCs, 2 Terminals, etc.). The wall plates are linked through horizontal wiring to administration points located in satellite closets.

For organizations requiring fiber optic connections, a multi-media wall plate is provided for unshielded twisted pair and fiber optic connections.

Intermediate Distribution Frames

To service large portions of office buildings (typically floors or major portions of floors) satellite wiring closets are established for increased flexibility and control. These wiring closets, or Intermediate Distribution Frames, tie together horizontal wiring from workstation locations on each floor to the backbone (riser) subsystem through the use of 110 type punch blocks.

Fiber optic cable in gathered in a similar fashion using fiber optic interconnection units for protection of the fiber and for easy patching of cable resources.

Riser Cable

This cable consists of high density, multi-pair wire (typically at least 300 pairs). This cable contains the primary circuit that connects Intermediate Distribution Frames to the Main Distribution Frame near the equipment location.

Fiber Optic riser cable is used to distribute fiber resources in much the same manner. These fibers are carried to the Main Distribution Frame also for interconnection to fiber optic electronics.

Main Distribution Frame

For connecting all remote IDFs, a Main Distribution Frame (MDF) is established. This will connect all remote wiring closets and thus all workstation locations into a signal central point. This collection of connections is called the "workstation" patching field within the MDF.

Unshielded Twisted Pair (UTP) Electronics

Most all technologies available in the industry today operate over unshielded twisted pair wire. For converting unbalanced media such as twinax, coaxial and other types of cable associated with IBM 3270, IBM 3X, Wang OIS, and Wang VS systems UTP devices are installed. These devices (baluns) convert unbalanced media to balanced UTP interfaces.

For local area network (LAN) connection over unshielded twisted pair cable, several manufacturers offer electronics for operating 3Com, Appletalk, Novell, Proteon, Ethernet, and 802.3 standards networks. These devices are widely available through companies like Hewlett-Packard and are many times more reliable than standard interfaces and offer superior troubleshooting capabilities and network management.

Resource

All information electronics such as PBX's, mini computers, PC LAN servers, Wang OIS, IBM mainframes etc. make resource subsystems. All resource components are connected to the MDF via fanout cables (and UTP devices if needed) and connect on 110 type punch blocks. This creates a "resource field" on the MDF representing the connection of all resource components to a single central point. To activate any resource to any workstation, then, a simple patch is made at the MDF. This process takes less than 30 seconds to complete.

So What?

Hopefully by now, I have convinced you that premises distribution systems fit together in a common architecture, seems flexible, are superior to coax cabling designs, and offers some fairly nice benefits. But lets take a look at some specific, tangible advantages of PDS and some real life benefits that these systems have provided.

Beyond the benefits of twisted pair versus coaxial cable that we discussed earlier, their are some very compelling reasons to implement PDS. These reasons include: 1) Increased cable management, 2) diagnostic capabilities, 3) increased response to to users, 4) disaster recovery, 5) configuration flexibility, and 6) reusability.

Increased Cable Management - With PDS, all of your organization's resources are accessible through 110 blocks. That is, any resource (computer, telephone switch, LAN, etc) can be connected to any wallplate (and thus a user) by a simple patch cord.

Documentation is improved through the use of special index strips located below each row of the 110 hardware. As moves and changes take place, the index strips easily identify existing locations and new or changed locations within the wiring closet. Simple documentation software applications are available for free from companies like The Apex Group. These systems identify individual relationships of user wallplates to company resources. That is, if you received a call from a user in Office 322, you could tell what type of telephone, workstation (and the specific configuration of those workstations) and what route the cable that services them travels, just by typing in their office number.

As moves take place, documenting these changes are very easily accomplished. Hardcopy reports can also be generated for a number of different network elements.

Enhanced Diagnostic Capability - Beyond the documentation systems described above, there are very sophisticated testing devices that work with PDS. By using the documentation system described above, you can identify the exact cable path servicing the user that is having the problem. Then, using testing devices that plug directly into standard PDS components (wallplates, 110 blocks, fiber optic interconnect boxes, etc.) you can identify the problem exactly or more importantly, determine that it is definitely not a cable problem. In this fashion, problems can be isolated in very short order.

Increased Response to Users - Another benefit directly related to improved diagnostics, and cable management is response time. Since problems can be isolated better, solutions can be provided faster. Our customers tell us us that their mean time to repair typically drops from 50 - 300% after implementation of a PDS.

Disaster Recovery - Disasters and accidents happen. PDS allows you to respond to these problems much faster. Since all components are compatible and modular, making emergency changes in response to an accident can be accomplished much easier.

One example of this disaster survivability occurred when a customer had one of his riser cables cut by maintenance worker thus severing the voice and data flow to an entire floor of his building. Through the use of his 110 patching fields, he re-routed this traffic to the floor below, and plugged into spare wire pair that lead to the computer room where the telephone switch and computers were located. Service was restored to the floor within 45 minutes. This work around was facilitated by the fact that all components used by every office were compatible, the entire system was modular, and all technologies used were connected over a common media - unshielded twisted pair cabling. Amazingly, the customer never even used a single tool and completed the work around through the patching and modular features of the cabling system.

Configuration Flexibility - With PDS you can treat your cabling system like a large matrix. Virtually, any resource can be connected to any wallplate or user. This connection can take place in less than a minute and can be changed as often as you wish. This gives you total flexibility and the ability to take advantage of new technologies as they are introduced. AT&T and Ungerman Bass have both announced their plans for equipment that operates at speeds of 100 Mbps over common twisted pair wiring.

Cable as an Asset not a Liability - Since nearly all technologies can take advantage of PDS, your cable plant can be a very valuable asset. If your organization ever moves out of its existing office or plant location, the new users will be able to use the PDS. You can mix and match PDS components to accommodate an entire new user community in very little time and for nearly no additional expense. When was the last time you remember that happening?

Let's take a look at some real life examples where PDS was installed and the benefits that were realized:

Example 1 - High Tech Fortune 50 Corporation

This customer was spending approximately $20,000 per month on moves, changes additions, and new cable runs. They had unusable documentation, and limited ability to troubleshoot problems.

A premise distribution system was installed for the price of $150,000. This system centralized all voice and data resources into a single distribution architecture. The were provided a cable management application for monitoring the movement of equipment, users, and connections throughout the facility, and complete hardcopy documentation and network maps.

New cabling run expenditures dropped off to less than $1,000 per month providing for a payback of less than 8 months. They immediately reported a 75% reduction in the number of problems they experienced in their user community, and reduced the time to respond to the remaining problems by 200%.

Moves and changes are administered by a secretarial-level person without tools or other involvement.

Example 2 - Large Diversified Construction Company

This corporation operated its corporate facility from a 4 building campus. Prior to the installation of a premises distribution system, they were spending $1,000 per month on local data circuits between buildings, they operated 4 separate PBX systems for each building, they had a high complaint rate of customers claiming they would get lost in the "maze", and they had distributed data centers. This corporation was spending approximately $10,000 per month on cabling plant changes and additions.

The decision was made to install a premises distribution system throughout the entire campus. This plant consisted of both copper and fiber optic components to each of the four buildings. The cost of installing this system was approximately $190,000.

The following benefits occurred:

1) They were able to consolidate phone system and realize lease savings, phone line savings, maintenance contract saving, and reduction in "operator" personnel to save approximately $9,000 per month.

2) While no formal measurement was taken before or after, customer complaint relative to being passed through the phone maze nearly disappeared.

3) Local data circuits between buildings were eliminated at a cost of $1,000 per month.

4) New cable runs were reduced to less than $500 per month.

5) Because data could now be distributed efficiently, multiple data centers were eliminated and consolidated into a single computer room. The company estimated that they realized $20,000 savings over the next 12 months, even after paying for moving charges.

6) The companies internal "hot-line" end-user support group was dismantled because the number of problem calls dropped substantially. This saving provided the company with a savings valued at over $100,000 in people that were re-assigned to more urgent projects within the MIS project.

The overall savings to the customer amounted to over $240,000 in raw dollars and greatly increased customer satisfaction. This provide this organization with a payback of less than 8 months.

Applications

Customer premise equipment can basically be divided into one of four major application areas. These applications are differentiated according to the size of the installation and the requirements for voice and/or data wiring.

Simple Wiring

Voice only wiring application, generally used for small businesses with one or several telephones that do not require separate wiring closets. Typically in buildings with less than 5,000 square feet of space.

Complex Wiring

Larger sites that require wiring closets and have extensive voice wiring requirements are described as complex wiring applications. This can include systems that are voice only or are integrated voice and data wiring schemes.

Structured Customer Premise Wiring

Structured Customer Premise Wiring is a total system wiring solution used primarily for integrated voice and data applications. Typical examples of this type of application include the systems installed by the major equipment manufacturers.

Data Wiring

Data only wiring for Local Area Networks such as coaxial based Ethernet, twisted pair based PC LAN's or terminal to host networks.

Besides the wiring application categories described above, building types and sizes will also effect the wiring system design. Basically, building types fit into one of six categories as follows:

Office Buildings
Manufacturing Buildings
Education
Health Care Facilities
Lodging
Mercantile/Service

When planning the design of an interconnection network, it is important to consider your individual requirements, including the application and building type. The good news is that designing these cable plants is relatively straight forward and costs usually $1,000 - $5,000. This can be done so reasonably due to the uniformity of PDS, and the design rules that are employed by qualified designers. Actual costs for facilities servicing over 500 offices may be slightly more.

Conclusion

Premises Distribution Systems can provide organizations with increased cable management, better diagnostic capabilities, configuration flexibility, disaster recovery, and reusable cable that becomes an asset for incoming occupants. Since nearly any voice or data technology can be used with PDS, substantial cost savings can be realized through the use of common unshielded twisted pair cabling, fiber optic cabling, and common interconnection components. This approach is substantially cheaper than mixed cable environments that use a combination of components and cable such as coaxial cable plants.

Moves, changes, and additions can become a minor occurrence. With a minimum of training, a secretarial level person can administer the entire cable plant for these changes. Reorganizations, office moves and other once disastrous events can be handled easily with a minimum of effort.

So the next time you feel like you are strapped to the side of a great stone mountain waiting for your next devouring, think of our old friend Prometheus and about Premises Distribution Systems for your organization.

Thank You.

Problem Isolation Techniques for TCP/IP Networks

Michael A. Pozzi
Hewlett-Packard Company
Colorado Telecommunications Division
5070 Centennial Boulevard
Colorado Springs, CO 80919

Introduction

Ethernet/802.3 local area networks using TCP/IP transport mechanisms tend to be large, multi-segment and multi-vendor. This complex networking environment typically consists not only of data transmission components, but of interacting systems, applications, subnets and external networks. Network management in this environment can be quite a challenge.

The goal of networking is to reliably and cost-effectively distribute information and resources among users. Very often the health and profitability of an enterprise can be significantly impacted by the efficiency and reliability of the network. This is particularly true as our dependence on information processing increases. Not only is it critical that the network remain operable and available, it must also be efficient and inexpensive. Users will demand good response times from networked applications. The business will require that it all be accomplished in a cost-effective manner.

The management of a TCP/IP based network must begin with an understanding of network operation under normal operating conditions. Only by knowing how the network behaves normally can one begin to use performance measurements to diagnose the problem when something goes wrong. Resource management and planning likewise depend on an understanding of network operation.

There exist several different tools to assist the network manager in these tasks. Network management systems available from third parties provide valuable information on the topology and operation of the network. Host-resident system diagnostics can be used to isolate problems in single vendor environments. But none of these tools is particularly effective when faced with the challenge of understanding, managing and optimizing the operation of a large, multi-vendor TCP/IP based Ethernet LAN.

This paper illustrates problem isolation techniques using the statistical performance analysis capabilities available on stand alone test tools: more specifically, protocol analyzers. In the past, the use of protocol analyzers has demanded significant levels of expertise on the part of the user in order to be effective. Analyzers which displayed pages after pages of decoded protocol messages were only valuable in the hands of a protocol guru. But recently, several of these products have been significantly enhanced for more novice users through the addition of powerful statistical measurements providing graphical interpretation of network performance information. No longer are network managers themselves required to deduce that information from files of collected protocol data. The use of these new problem solving techniques will be illustrated with example problems.

Common Problems

Most of the networking problems which commonly plague managers of TCP/IP networks fall into four categories: hardware problems, addressing problems, configuration problems and network congestion. In order to best illustrate techniques for dealing with these problems, we will look at one example from each category and describe the problem isolation procedure.

Though the focus of this paper is on problem solving, it is worth noting that these measurement techniques can be equally well applied to network planning and preventative maintenance. Following the case studies we will describe both planning and problem prevention techniques using many of the same measurements.

Hardware Problems

Ethernet cables, connectors and transceiver circuits are all susceptable to failure. In some cases the failure modes are catastrophic and relatively easy to diagnose. A severed cable, for example, is quickly isolated with an electrical continuity check, with one of several commercially available cable testing devices, or in some cases even a TDR (time domain reflectometer). The more difficult hardware problems are not catastrophic, however, but rather intermittent and much more subtle. Isolating intermittent problems can be very difficult, especially if the failures are infrequent.

The traditional approach to isolating an intermittent problem is to capture data continuously until the problem occurs. This process can be very tedious and time consuming. Sophisticated triggering schemes based on pattern matches within protocol frames are of little use, since hardware failures can corrupt data in random ways.

Some protocol analyzers are equipped with the ability to statistically measure the levels of errored frames over time. These errors include misaligned frames or frames with invalid frame check sequences, frame fragments (runts), extremely long frames (jabbers) and number of collisions. Some number of these errors is normal for most networks. Knowing what levels of errors to expect, the network manager can set threshold levels for each or all of the errors so that any measurement of error occurance above the set thresholds will trigger an alarm.

The key to quick isolation of intermittent problems is not only the ability to alarm on errored frames, but also the ability to program the protocol analyzer to automatically run a pre-defined list of tests whenever the alarm occurs. These tests, which are also statistical measurements, are designed to determine exactly which node on the network is responsible for the large number of errored frames. The example of figures 1 through 4 illustrates this technique.

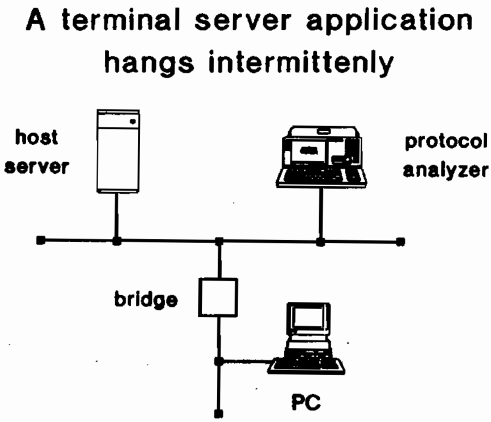


Figure 1 In this example, the PC's connection to the host goes dead or "hangs" intermittently at random times during the day.

```
                              SET ALARMS
10 Jul 89                                                     00:02:34
        Network Stats                           Node Stats
 Utilization:                         Connect. Stats:    Status  Maximum
   Status    Minimum      Maximum      # Connections:    Off        99
    Off       0.00 %      50.00 %     Qualify alarm for:   1 sample(s)
    Off       0.00 f/S     7000 f/S
    Off       0.00 kB/S    5000 kB/S          Transmit Stats
  Qualify alarm for:   1 sample(s)   Channel Acquis:    Status  Maximum
                                       Acquisition Time: Off    100000 uS
 Errors/Coll:  Status     Maximum      % Msgs. Deferred: Off     50.00 %
   All errors : On   1.00E-03 err/frm  Collisions/Msg. : Off      8.00
   FCS/Misalgn: Off  1.00E-03 err/frm  % Msgs. Aborted : Off      0.00 %
   Runts      : Off  1.00E-03 err/frm Qualify alarm for:   1 sample(s)
   Jabbers    : Off  1.00E-10 err/frm
   Collisions : Off  1.00E-01 col/frm Response Time:    Status  Maximum
  Qualify alarm for:   1 sample(s)     Response Time   : Off     50000 mS
                                      Qualify alarm for:   1 sample(s)

 Alarm type = Audible                          Alarm duration =    5 Seconds
```

Figure 2 The alarm threshold is set at 1 frame in 1000 for all types of errors in any one second time interval.

```
                           AUTOMATIC SEQUENCE
10 Jul 89                                                     00:01:35

 SEQ.  MEAS    MEASUREMENT        MEAS. TIME   ALARM  ELSE       COMMENTS
  #    CLASS   OR FUNCTION        OR SETTING   GO TO  GO TO
 ----  ----   ---------------     -----------   ---    ---   -----------------
   1.  NET    NETWORK SUMMARY      60 Seconds    2      1
   2.  NODE   NODELIST STATS       30 Seconds   N/A     3
   3.  CTRL   END SEQUENCE            N/A       N/A    N/A   End of sequence.

Start time = 10 Jul 89  23:44:33                     Start at sequence #    1
```

Figure 3 The instrument is programmed to run the node list statistics whenever the alarm threshold is exceeded.

NODE LIST STATISTICS

10 Jul 89 01:39:38

Node #	Node Name or Address	Last Sample Frame	Cnt.	Frame Count	KByte Count	Error Count	Error Rate	Avg.Frm Size
1	MO_KIRWAN	XMT	93	124	7	90	7.26E-1	46
2	FINANCE_SERVER	XMT	162	216	22	0	0.00E+0	91
3	LAB_HOST	XMT	117	156	12	0	0.00E+0	64
4	MKTG_SERVER	XMT	92	124	9	0	0.00E+0	64
5	WANDA_PC	XMT	63	84	6	0	0.00E+0	60
6	TERM_SERVER	XMT	45	60	5	0	0.00E+0	67

Start time = 10 Jul 89 01:33:25 Stop time = 10 Jul 89 01:33:26
Measurement time = 1 Second Sample time = 1 Second

Figure 4 The node list statistics quickly point to the offending network node, which in this example produced over 70% errored frames in a one second period.

The faulty component in this case is the transceiver card in the node labeled "MO_KIRWAN". Replacement of this interface quickly restores network operations.

Addressing Problems

Addressing problems on TCP/IP networks can manifest themselves in many ways. One of the most common symptoms is illustrated in Figure 5. Two stations are unable to communicate over a network when at the same time none of the other stations report any difficulties.

Computer Museum

A user is unable to read his electronic mail when communication with a distant host fails

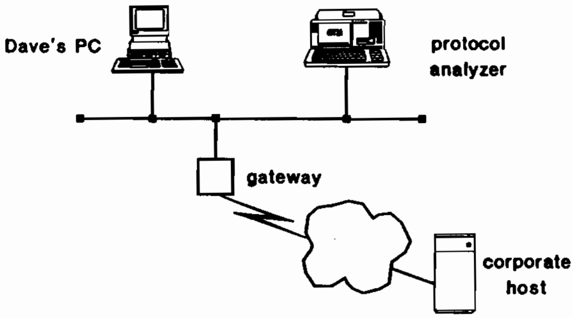


Figure 5 Dave's PC is unable to establish a session with the remote corporate host. All of the other workstations on Dave's segment can access the host with no problem.

Suspecting an addressing problem, the network manager might begin the problem isolation process by verifying the IP address of Dave's PC. This is easily accomplished with a protocol analyzer with stimulus/response testing capabilities. The analyzer is programmed to transmit an address resolution protocol (ARP) message, which is a procedure built into the TCP/IP protocols for the purpose of resolving addressing problems. The ARP message is broadcast throughout the network, searching for the physical address of the station with a given IP address.

```
Listing of program "TRY_ARP" :

Store: frames matching ARP_REQUEST
        or ARP_REPLY
       until full

Log file: not used

Block 1:
     Send message ARP_REQ
       and then
     Wait    2 seconds
       and then
     Stop test
```

Figure 6 The protocol analyzer comes pre-programmed to broadcast an ARP message and look for responses. The network manager need only enter the IP address of the station in question.

```
#1         Jul 16 @23:47:09.27282 Len   60 Filters 0xxx............       No error
Ether: Dst BROADCAST            Src HP_LAN_Analyzer          Type                ARP
ARP  : Hardware                 Ethernet (10Mb)   Protocol                    DOD_IP
     : Hardware Addr Len                6 Bytes   Protocol Addr Len          4 Bytes
     : Operation              1    ARP Request
     : Sender Host              HP_LAN_Analyzer   Sender Internet            HP_4972A
     : Target Host            00-00-00-00-00-00   Target Internet           dave_bass

#2         Jul 16 @23:47:09.33160 Len   60 Filters x1xx............       No error
Ether: Dst HP_LAN_Analyzer      Src JOE                      Type                ARP
ARP  : Hardware                 Ethernet (10Mb)   Protocol                    DOD_IP
     : Hardware Addr Len                6 Bytes   Protocol Addr Len          4 Bytes
     : Operation              2  ARP Response
     : Sender Host              HP_LAN_Analyzer   Sender Internet            HP_4972A
     : Target Host                          JOE   Target Internet           dave_bass

#3         Jul 16 @23:47:09.43186 Len   60 Filters x1xx............       No error
Ether: Dst HP_LAN_Analyzer      Src DAVE_BASS                Type                ARP
ARP  : Hardware                 Ethernet (10Mb)   Protocol                    DOD_IP
     : Hardware Addr Len                6 Bytes   Protocol Addr Len          4 Bytes
     : Operation              2  ARP Response
     : Sender Host              HP_LAN_Analyzer   Sender Internet            HP_4972A
     : Target Host                    DAVE_BASS   Target Internet           dave_bass
```

Figure 7 Immediately after transmitting the ARP message, the analyzer receives two responses. There are two physical stations on the network with the sane (duplicate) IP address.

In this case, the analyzer quickly isolates the problem: two physical stations on the network are both using the same IP address. Duplicate IP addresses are in fact a common problem on TCP/IP networks. This condition can be created any time a program or setup file is copied from one workstation to another. In this case, the multiple occurance of the same IP address is confusing the gateway, which must route traffic to the correct physical node based on a mapping of IP address to physical address. Traffic from the corporate host destined for Dave's workstation is being forwarded to the correct IP address, but the wrong physical station. The solution is to eliminate any duplication of IP addresses.

It is important to choose a protocol analyzer capable of stimulus/response testing. Many analyzers lack the flexibility to program the transmitter, and some are incapable of monitoring network responses while transmitting. It turns out that many classes of problems can be quickly isolated by running simple pre-written programs like the ARP message.

Configuration Problems

Because the TCP/IP networking environment will very often contain devices from several different vendors, it is somewhat prone to problems resulting from device incompatibles or configuration mismatch. One common example, illustrated in the network diagram of figure 8, involves data flow control between a file server and a workstation.

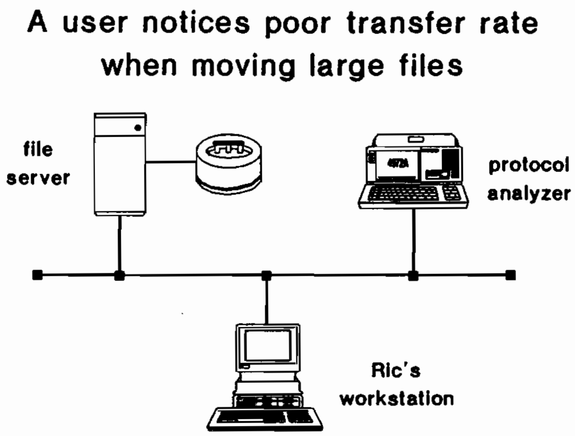


Figure 8 Large files delivered from the file server to the workstation are transferred extremely slow, despite the fact that the Ethernet network itself has plenty of spare bandwidth.

In order to zoom in on this particular file transfer process, the protocol analyzer must be capable of picking out one particular conversation from the multitude of traffic present on the Ethernet cable. Solving this problem quickly will require an analyzer be capable of doing statistical measurements which relate only to that particular transaction. With the recent advances in testing technology, both requirements can be met.

Figure 9 illustrates a throughput measurement on the file transfer process in question. It can be seen at a glance that the data throughput for this connection degrades significantly as the transfer proceeds. Viewing the same information in a tabular format (Figure 10) reveals the connection to be very clean, with no packet retransmissions. It also reveals, however, a significant decrease in data packet size over the course of the transfer.

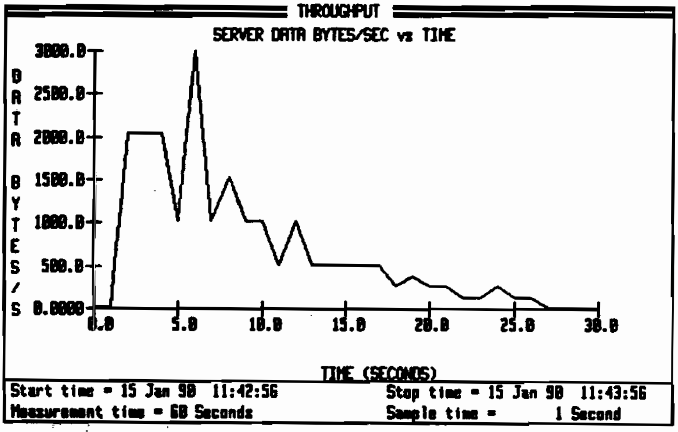


Figure 9 Data throughput for a particular connection can be extracted from Ethernet traffic and displayed graphically.

THROUGHPUT

6 Feb 90 02:40:04

SERVER IP: mktg_server CLIENT IP: ric_bechter

PORT: FTP_DATA INITIAL PORT: AUTOMATIC

ELAPSED SAMPLE TIME	CLIENT PORT	SOURCE	TOTAL PACKETS	DATA PACKETS	DATA BYTES/S	DUPLICATE PACKETS	AVERAGE TIMEOUT
00:00:18	6168	client	2	0	0	0	
		server	2	2	512	0	
00:00:19	6168	client	2	0	0	0	
		server	2	2	256	0	
00:00:20	6168	client	2	0	0	0	
		server	2	2	256	0	
00:00:21	6168	client	1	0	0	0	
		server	1	1	128	0	
00:00:22	6168	client	2	0	0	0	
		server	1	1	128	0	

Start time = 6 Feb 90 02:34:33 Stop time = 6 Feb 90 02:35:33

Measurement time = 60 Seconds Sample time = 1 Second

Figure 10 Zero duplicate packets indicates a clean connection with no retransmission. However, decreasing packet sizes may indicate a flow control problem.

Data flow control on TCP/IP networks is handled by a mechanism known as "send window". The send window is a number sent from the receiving station to the transmitting station on each acknowledgement message during the course of a file transfer. The send window advises the transmitting station of the amount of buffer space remaining in the receiving station. If the transmitter can send data faster than the receiver can process, then the send window value will decrease over the course of the file transfer. This condition is evident in the measurement shown in Figure 11.

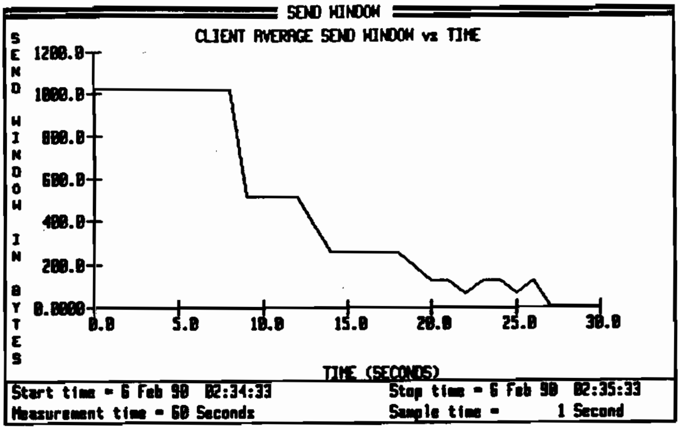


Figure 11 The measurement of send window size indicates that the input buffer of the receiving station is filling rapidly, as data is coming in faster than the workstation can process it.

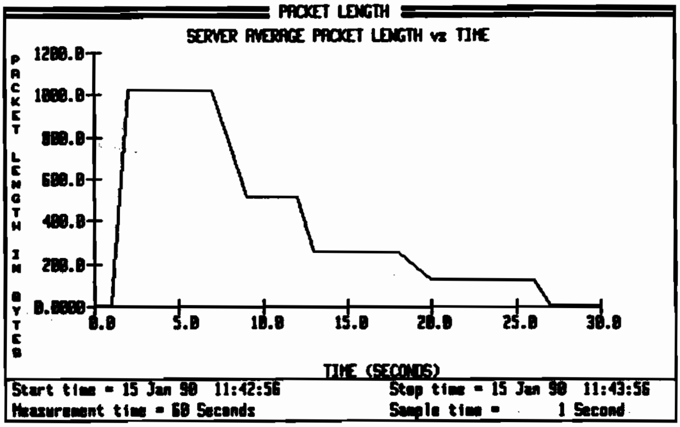


Figure 12 The constricting send window is causing the file server to transmit smaller and smaller packets.

If the file server responds to this condition of constricting window size by transmitting smaller and smaller packets, eventually the overhead associated with the transmission of each packet through the network will make the process extremely inefficient. That is exactly the cause of the degrading throughput in this case. This phenomena occurs so frequently that it is commonly referred to as the "silly window syndrome".

One solution to this problem is for the workstation to refuse to offer any non-zero window size until it can accept a reasonably sized packet. The server may also ignore the send window and refrain from sending data until the window opens up to at least 25% of its original value. Many vendors have revised their applications to avoid this condition using one or both methods.

It is worth noting the ease with which such configuration problems can be isolated using statistical measurements presented in graphical format. Though it is possible for a knowledgeable protocol expert to extract the same results from the examination of captured data messages, that process would be far more tedious and time consuming. It is important to select test tools with statistical measurement capabilities.

Network Congestion

Devices spanning Ethernet segments, such as bridges, routers or gateways can become overloaded at times of peak traffic. When that happens, these devices will begin to "drop packets". That is, some percentage of the traffic will simply not make it through the device. Unless the network manager is aware of each device's rated capacity relative to the actual traffic load, it can be very difficult to isolate such problems.

In the example of Figure 13, users on one segment of a large network report intolerable delays and poor response when communicating with a distant host. The network manager's first step in resolving this reported problem might be to measure the data throughput and the reponse time for one of the connections in question. Such measurements again will require the protocol analyzer to pick out a single conversation from the traffic stream and measure some key performance statistics on that connection.

Users on one subnet report intolerable delays when communicating with distant host

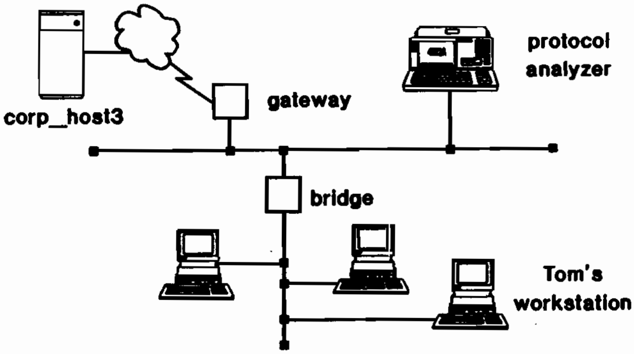


Figure 13   Protocol analyzers can be used to isolate a single conversation from the complex data stream on an Ethernet backbone.

The analyzer is connected to the Ethernet backbone network and programmed to capture a conversation between one of the subnet workstations and the remote host. Note that this can only be done by filtering the conversation based on the IP addresses and TCP ports. Filtering solely on physical addresses would not be enough to isolate this particular transaction. The results are the throughput and response time measurements of figures 14 and 15. In this case, packets are being lost and retransmitted, causing delays in response of more than three seconds.

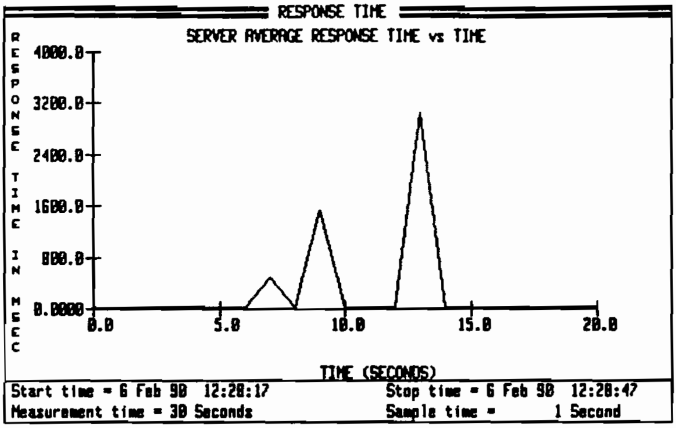


Figure 14 Response times of greater than three seconds are measured on the conversation between the remote host and one of the subnet workstations.

THROUGHPUT

1 Apr 90 00:11:23

SERVER IP: CTD_HOST CLIENT IP: CTD_WORKSTATION

PORT: FTP_DATA INITIAL PORT: AUTOMATIC

ELAPSED SAMPLE TIME	CLIENT PORT	SOURCE	TOTAL PACKETS	DATA PACKETS	DATA BYTES/S	DUPLICATE PACKETS	AVERAGE TIMEOUT
00:00:14	7749	client	3	3	1,535	1	1.01
		server	1	0	0	0	
00:00:15	7749	client	0	0	0	0	
		server	0	0	0	0	
00:00:16	7749	client	2	0	0	2	1.51
		server	1	0	0	0	
00:00:17	7749	client	0	0	0	0	
		server	0	0	0	0	
00:00:18	7749	client	1	0	0	1	2.01
		server	0	0	0	0	

Start time = 10 Jul 89 16:36:22 Stop time = 10 Jul 89 16:37:22

Measurement time = 60 Seconds Sample time = 1 Second

Figure 15 Delays in data throughput are being caused by lost packets. The two columns farthest right report the number of packets retransmitted and the average timeout for each one second sample period.

Though response time is a very fundamental and straightforward concept, it is not at all an easy thing to measure. Protocol analyzers which are designed solely to monitor and decode protocol frames are often unable to make a response time measurement. When faced with large files of captured protocol frames, the network manager is forced to search through those frames looking for a matching pair of data and acknowledgement messages. In TCP/IP, this means keeping track of data sequence numbers, packet lengths and acknowledgement numbers. Users need also be mindful of packet retransmissions, as the true reponse time is measured from the first transmission of data to the eventual acknowledgement, regardless of the number of intervening retransmissions. Clearly it is more desirable to use an analyzer capable of automatically measuring and graphing response time for any conversation defined by an IP address and TCP port pair.

```
#5          Jan 23 @ 0:07:49.00354 Len  566 Filters x...............     No error
Ether: Dst ctd_gateway         Src tom_x4278              Type              DOD_IP
IP   : Src            tom_smith Dest       15.6.72.102 Protocol                TCP
TCP  : Src                 7748 Dest     FTP_DATA Sequence Number        169288267
     : Ack          29733698 Data Offset          20 Flags                 ACK PSH
     : Window              1024 Cksum     Good  9C-5A Urgent Pointer   Not Used    0
     : Direction           Client --------------> Server    ( 512 bytes )

#6          Jan 23 @ 0:07:49.00866 Len  566 Filters x...............     No error
Ether: Dst ctd_gateway         Src tom_x4278              Type              DOD_IP
IP   : Src            tom_smith Dest       15.6.72.102 Protocol                TCP
TCP  : Src                 7748 Dest     FTP_DATA Sequence Number        169288779
     : Ack          29733698 Data Offset          20 Flags                 ACK PSH
     : Window              1024 Cksum     Good  A6-35 Urgent Pointer   Not Used    0
     : Direction           Client --------------> Server    ( 512 bytes )

#7          Jan 23 @ 0:07:49.00914 Len   60 Filters x...............     No error
Ether: Dst tom_x4278           Src ctd_gateway            Type              DOD_IP
IP   : Src         15.6.72.102 Dest          tom_smith Protocol                TCP
TCP  : Src             FTP_DATA Dest             7748 Sequence Number      29733698
     : Ack         169288779 Data Offset          20 Flags                     ACK
     : Window              7168 Cksum     Good  E3-69 Urgent Pointer   Not Used    0
     : Direction           Server --------------> Client    ( 0 bytes )
```

Figure 16 Response time measurements can be derived from decoded protocol frames, but it is a tedious and time consuming process. It is far more desirable to choose an analyzer capable of making response time measurements making response time measurements automatically.

The next step in our example must be to isolate the cause of the delay. By measuring response time on this same conversation at the remote host connection, the response time delay of the host can be separated from potential delays through the wide area network. Doing this measurement reveals that the host itself is not the cause of the problem. (Figure 17)

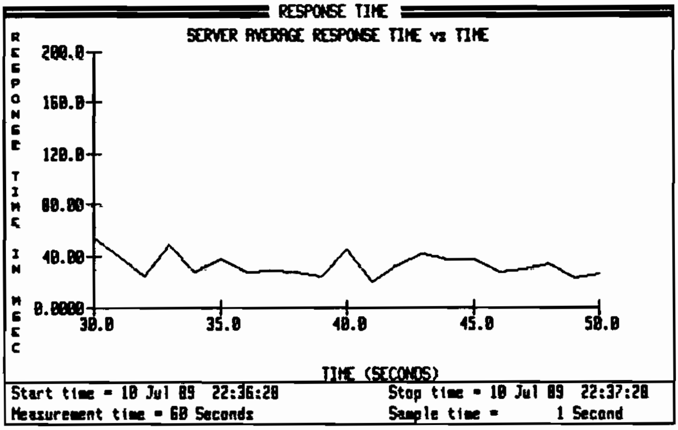


Figure 17 The actual reponse time of the host itself is not a significant factor in this case. This measurement must be made at the remote site, at the host connection to the wide area network.

Measurement of the volume of traffic through the gateway can be made by first filtering to see only those packets to or from the physical address of the gateway, and then running statistics on individual nodes by IP address (Figure 18). This measurement reveals that over 21,000 packets were transmitted through the gateway in the last eleven seconds to the remote host connection. This traffic volume is approaching the specified gateway capacity of 2,000 packets per second, and may be resulting in the loss of data packets by that device. The solution to this problem would be either to install a second or more powerful gateway, or perhaps to reduce the overall volume of interactive traffic through the gateway destined for the host by installing a dedicated server on the local subnet.

IP ADDRESS LIST STATISTICS

1 Apr 90 00:12:31

Node #	Node Name or Address	Dir	--Frame Counts--- Sample	Total	KByte Count	Avg FrSz	Last Transmission
1	corporate_host	XMT	1,876	21,397	838	40	17 Jul 00:18:09
		RCV	1,875	21,397	483	23	
2	tom_smith	XMT	235	2,675	52	20	17 Jul 00:18:09
		RCV	470	5,350	329	63	
3	mike_y	XMT	469	5,349	104	20	17 Jul 00:18:09
		RCV	235	2,675	89	34	
4	dave_n	XMT	234	2,674	81	31	17 Jul 00:18:09
		RCV	468	5,348	115	22	
5	mike_b	XMT	468	5,350	141	27	17 Jul 00:18:09
		RCV	234	2,675	138	53	
6	david_b	XMT	235	2,675	52	20	17 Jul 00:18:09
		RCV	235	2,675	52	20	

Start time = 17 Jul 89 00:17:58 Stop time = 17 Jul 89 00:18:09
Measurement time = 11 Seconds Sample time = 1 Second

Figure 18 Traffic volume can be measured by isolating packets on the backbone coming to or from the physical address of the gateway. The traffic through the gateway to and from the remote host alone exceeds the 2,000 pps capacity of the gateway.

Network Planning and Preventative Maintenance

The same statistical performance measurements which are so helpful for problem isolation are also extremely valuable for network planning and preventative maintenance. Knowing who, what, when and how the network is being used enables you to better understand network operation. And it is only through understanding the operation of the network that you can hope to manage it.

NODE LIST STATISTICS

6 Feb 90 13:10:03

Node #	Node Name or Address	Last Sample Frame Cnt.		Frame Count	KByte Count	Error Count	Error Rate	Avg.Frm Size
1	R&D_ROUTER	XMT	1,208	1,208	110	302	2.50E-1	79
		RCV	1,208	1,208	97	302	2.50E-1	69
2	FINANCE_GATEWAY	XMT	1,208	1,232	99	308	2.50E-1	69
		RCV	906	924	90	0	0.00E+0	85
3	MKTG_SERVER	XMT	906	942	114	0	0.00E+0	109
		RCV	604	628	237	0	0.00E+0	366
4	DAVE_WKSTN	XMT	302	320	51	0	0.00E+0	148
		RCV	302	320	48	0	0.00E+0	138
5	WANDA_PC	XMT	302	320	191	0	0.00E+0	584
		RCV	302	320	37	0	0.00E+0	104
6	TERMINAL_SERVER	XMT	0	0	0	0	0.00E+0	0
		RCV	604	652	48	0	0.00E+0	62

Start time = 6 Feb 90 13:09:55 Stop time = 6 Feb 90 13:10:20
Measurement time = 25 Seconds Sample time = 5 Seconds

Figure 19 **Node list statistics, captured either by physical or by IP address, reveal the amount of traffic generate by each station or each application. This is the measurement of "who" is communicating over the network.**

TCP/IP NETWORK SUMMARY

6 Feb 90 13:18:51

Utilization

Current	Average	Peak	
9	9	9	total f/s
2	2	2	IP f/s
53	53	53	IP bytes/s

IP Frame Parameters

Total data bytes	3,183
Total frames	109
Total fragments	0
Header overhead	40.65 %

TCP/UDP Port Activity

	Current	Average	Peak
FTP	3.67	3.67	3.67 %
FTP_DATA	0.0000	0.0000	0.0000 %
TELNET	88.99	88.99	88.99 %
SMTP	0.0000	0.0000	0.0000 %
SUN_NFS	0.0000	0.0000	0.0000 %
SUN_PC_N	0.0000	0.0000	0.0000 %
RWHO_RLO	1.83	1.83	1.83 %
OTHER	5.50	5.50	5.50 %

Start time = 6 Feb 90 13:17:36 Stop time = 6 Feb 90 13:18:36
Measurement time = 60 Seconds Sample time = 1 Second

Figure 20 **Traffic distribution by TCP port indicates "what" is being communicated over the network. (File transfers, interactive terminal traffic, workstation communication, etc.)**

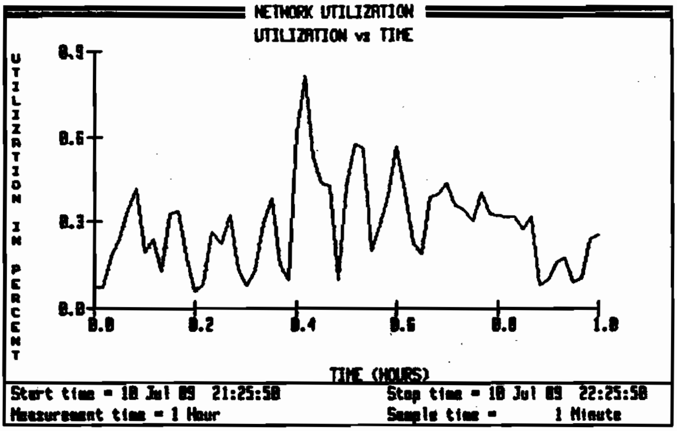


Figure 21 Total traffic volume expressed as a percentage of the available network bandwidth is known as utilization. Essential for capacity planning, this measurement reveals "when" the network is being used.

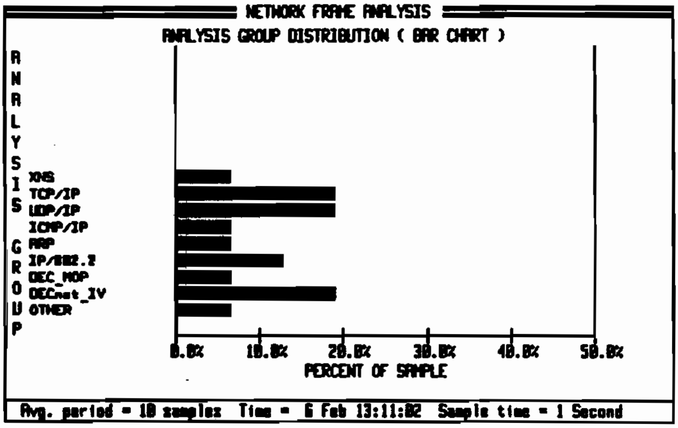


Figure 22 Network traffic can also be analyzed by the transport protocols being used, such as TCP/IP, XNS or DECnet.
This measurement reveals "how" communication is being accomplished.

Conclusion

The understanding of network operation is essential for quick problem isolation. This means measuring the throughput of individual devices, knowing the response times and data packet sizes for a specific connection, understanding usage patterns for network applications over time, and knowing which network resources are being consumed by which users.

Protocol analyzers which make these statistical performance analysis measurements can be extremely valuable tools for problem isolation. Without requiring a great deal of detailed protocol expertise of the user, these tools can quickly reveal the efficiency of a particular conversation, data throughput rates, network response times and traffic volume. Such information can quickly isolate the most common problems which typically plague a TCP/IP based network. They can be very valuable as well in network planning and in preventative maintenance.

Program to Program Communication HP and IBM

F. Allan Gregory
E.I. du Pont de Nemours and Co., Inc.
Wilmington, Delaware

ACKNOWLEDGEMENT

George Koffenberger and Joe Szaflarski of E. I. du Pont de Nemours and Co., Inc. were responsible for much of the design, coding techniques, testing and implementation of the LU6.2 utility described in this paper.

SUMMARY

HP's product offering of the IBM SNA Advanced Program-to-Program Communication facility (APPC) allows computerized business transactions to flow quickly to or from applications on an IBM system from or to applications on an HP system.

By implementing APPC as a data communications utility which is callable at the application level, different applications can use the same communications software thereby reducing development and implementation time and cost. In addition, this loose coupling of applications across systems permits rapid data flow when all systems and applications are available, but does not require that an outage on one system affect the other.

BACKGROUND

In late 1988, a division of DuPont was beginning implementation of manufacturing applications in support of a just-in-time manufacturing strategy. The division would be using HP systems at plant sites for the manufacturing applications, but customer orders would be going into a centralized corporate order processing system on IBM equipment. When all systems were in place in late 1989, the customer orders would need to be sent from the order processing system on the IBM to the manufacturing system on the HP.

The existing data transport available from IBM to HP was Remote Job Entry. This post-processing and batching of orders did not seem to fit well with the philosophy of just-in-time manufacturing nor with the transaction-oriented applications on both the IBM and HP.

The other alternative was the HP emulation of the IBM APPC protocol LU6.2. Testing and development were started with LU6.2 on the HP3000. The LU6.2 utility that resulted from this development was initially installed in late 1989. The same software has since been installed unchanged at other sites for other divisions, each of which runs different applications software on their HP systems.

LU6.2

LU6.2 is a conversational protocol which allows one application program to "talk" with another application program over SNA links.

Systems Network Architecture (SNA) has defined device connectivity for IBM for 15 years. Within SNA both physical units (PUs)--hardware--and logical units (LUs)--software--are defined. There are various types and subtypes of hardware and software (PUs and LUs) each with their own corresponding capabilities and protocols. Logical unit type 6, subtype 2 (LU6.2) provides the definition of capabilities and protocols necessary for application program-to-program data flow over an SNA link.

The HP product SNA Link (30246A or 30291A) provides the hardware path and emulation and control software necessary for the HP3000 to be defined and functional in an SNA network as a Physical Unit Type 2.0 cluster controller capable of LU6.2 communications. The products include either an INP or PSI board as well as software. The HP3000 is physically attached to an IBM communications controller on the SNA network (typically via a modem and leased line).

The HP product LU6.2 Application Programming Interface (30253A/R or 30294A/R) is a set of intrinsics that provide the capability for high level application programs to participate in an LU6.2 conversation. The API is currently (May, 1990) callable from COBOL II, Transact, Pascal or C (XL only).

LU6.2 CONVERSATIONS

LU6.2 is a "conversational" protocol between two programs. It can be very much analogous to a telephone conversation:

----Mary----	----John----
Hello. Is John there?	
	Yes, this is John.
Tomorrow's lunch will be at 12:30.	
	OK, got it.
There will be 5 of us.	
	OK, got it.
That's all.	
	OK. Goodbye.
Goodbye.	

The LU6.2 protocol consists of a number of verbs, each with its own function. For example, ALLOCATE will start a program on another system; SENDDATA does as its name implies as does RECEIVEANDWAIT. These verbs are accessed as intrinsics on the HP3000. They are used by application programs in a manner very much like the telephone conversation:

SENDING PROGRAM ----Mary----	RECEIVING PROGRAM ----John----
ALLOCATE Receiving Program Hello. Is John there?	GETALLOCATE Yes, this is John.
SENDDATA Tomorrow's lunch will be at 12:30.	RCVANDWAIT OK, got it.
SENDDATA There will be 5 of us.	RCVANDWAIT OK, got it.
CONFIRM That's all.	CONFIRMED OK.
	DEALLOCATE
DEALLOCATE Goodbye.	Goodbye.

There are, of course, parameters that must be supplied to each intrinsic. A typical call on the HP would be:

```
CALL INTRINSIC "ALLOCATE" USING TP-ID, SESSION-TYPE, REMOTE-PROGRAM-NAME,
     REMOTE-PROGRAM-NAME-LENGTH, ALLOCATE-ID, STATUS, SYNC-LEVEL.
```

Each parameter is described in the manuals that accompany the HP products. There are status codes and error conditions that must be checked for each call.

The major LU6.2 verbs available are:

ALLOCATE	Send a request to start a conversation.
CONFIRM	Send a confirmation request and wait for a reply.
CONFIRMED	Send a confirming reply in response to a confirm request.
DEALLOCATE	Stop a conversation.
GETALLOCATE	Receive a request to start a conversation and do so.
GETATTR	Obtain information about a conversation.
PREPTORCV	Announce that you are ready to receive.
RCVANDWAIT	Wait for information to arrive. A calling parameter must be tested on return to see what was received--data, control information, or confirmation request.
REQTOSEND	Announce that you would like to send.
SENDDATA	Send data.

Conversations may be in various states (send or receive for example) depending on what verbs were just executed and what the results were. Certain verbs may only be called in certain states. Each program in conjunction with its conversational partner must manage the form or style of the conversation as well as the status. The order and use of the LU6.2 verbs is totally up to the conversational partner programs.

Conversation styles may be constructed in any way that is mutually agreeable to the two parties. Data may be used to turn the conversation around or to terminate it. One program is not limited to receive or send only. The verbs and their use constitute a conversation based partnership.

LU6.2 CHARACTERISTICS

Some characteristics of LU6.2 conversations are:

* Processor Neutral

  Whether an HP or an IBM program initiates the conversation makes no difference. Either can start a conversation and either can send or receive or both depending on what conversation flow both programs adhere to.

* Context Dependent Partnership

  It is imperative that the program-to-program communication be viewed as a partnership. Conversation styles and flows must be established and followed. Designing in a vacuum will not work.

* Fairly Simple, Common Sense

  The LU6.2 verb set and their use will be seen to be fairly straight forward once some experience with them is established and once some basics of conversation state and status and control codes are understood. By far the best way to become fluent is to get a partner, set up an environment, and start "talking".

LU6.2 UTILITY

The LU6.2 conversation above (John and Mary) could be shown in a block diagram as:

```
                           SNA
          _____________          _____________
         |  SENDING    |<-----/  |  RECEIVING  |
         |APPLICATION  |     /   |APPLICATION  |
         |_____________|    /--->|_____________|
                          LINK
```

In this implementation:

* the LU6.2 conversation verbs are in the application
* the applications are tightly coupled; what one does is dependent on the other
* conversations are tailor-made by application
* changes in data communications have to be reflected in application changes.

All of the above can be addressed in various ways and for some applications might be features.

A compounding reality in the case at DuPont is that the major IBM application involved (corporate order processing) has to deal with many applications on many technologies. The solution was to design and develop a standard LU6.2 conversation on the IBM that could be invoked when a remote system needed to send data and to define a standard LU6.2 conversation that the IBM would use to send data to a remote system.

In implementing the LU6.2 capability for the HP technology, it was decided to package access to these standard conversations as application level callable utility subroutines. In practical terms, there are two subroutines available to the applications programmer--one to call if the application has data it wants to send and one to call to see if any data has come in. Although implemented through system services, the function on the IBM system is much the same. In a block diagram:

```
                                 SNA
   ____        ____  <------/          ____        ____
  |    |CALL  |L U |       /          |L U |CALL  |    |
  |APPL|<---->|U T |      /           |U T |<---->|APPL|
  |    |      |6 I |     /            |6 I |      |    |
  |    |      |2 L |    /             |2 L |      |    |
  |____|      |____|   /--------->    |____|      |____|
    HP          HP    LINK              IBM         IBM
```

Now, rather than learning and calling individual LU6.2 intrincics, the HP application programmer need only make one of two calls:

```
CALL SENDLU62 USING DESTINATION, DATA-LENGTH, DATA, STATUS.

CALL RECVLU62 USING SOURCE, DATA-LENGTH, DATA, STATUS.
```

All of the calling of LU6.2 verbs, the management of the conversation, data formatting (data coming from or going to IBM must be translated to EBCDIC), status and error checking are done by the LU6.2 utility.

LU6.2 UTILITY OPERATION

When the application calls the SENDLU62 subroutine, the data is not actually sent at that time. Instead, the transaction is written to an intermediate disc file. It is then read by a continuously running background job that actually does the LU6.2 communication.

On the receiving side, jobs are executed on the HP whenever the IBM does an LU6.2 ALLOCATE (this is a feature of the HP emulation). Data is received by that job and written to an intermediate disc file. A continuously running background job containing application level code calling the RECVLU62 utility subroutine will constantly check that file for the incoming data.

All data sent or received is logged to disc. Utilities are available to list what has been sent or received and to retransmit if necessary.

The use of message files for transactions sent or received and circular files for log files eliminates the need for file maintenance activties.

Because the LU6.2 is independent from the application and intermediate files are used, transactions being sent from the application will queue if either the communications or the receiving application is not available and data coming from the IBM application will queue if the HP application is not available. Data will queue on the IBM if the HP system or SNA is not available.

Although the intermediate presence of the utility and the disc writes do add to the overhead of data transmission, performance is quite acceptable. When sitting at side by side terminals, one going into the IBM application and one going into the HP application, the result of transactions entered in one application can be viewed in the other system in less than two seconds for most transactions. It should be noted that this is at low to moderate transaction volumes. High transaction volumes might result in different performance and dictate some different designs.

LU6.2 UTILITY ADVANTAGES

There are several advantages in implementing the LU6.2 communications as an application callable utility:

* Applications programmers do not have to learn the LU6.2 protocol. This obviously reduces development time and cost.

* More rapid deployment of the technology because the learning curve is small.

* Backup commuications are easily put in place. (It is very easy for example to replace a batch job doing LU6.2 data communications with a batch job doing RJE communications totally transparent to the applications queueing transactions).

* Changes in data communications do not effect application programs.

* The same utility software is available to many applications at many locations.

* Fewer problems and easier and lower-cost maintenance because of one solution versus many.

LU6.2 UTILITY DISADVANTAGES

Because of the conversation style implemented and to some extent because of the nature of the transactions of the applications involved, the LU6.2 utility program-to-program communications is more of an event-driven single transaction data transmission facility rather than a true cooperative processing environment. The difference is that there is no immediate application level feedback as part of the communications. Error reporting is handled separately from the transaction itself. This can make some error tracing difficult. Although this is primarily due to the nature of the transactions themselves, an intermediate utility could affect this capability. In order to keep the utility concept and still do effective cooperative processing, either a different LU6.2 utility conversation would need to be designed or application transactions would need to be defined in a way that more closely coupled the applications.

CONCLUSION AND OBSERVATIONS

Overall the experience with LU6.2 on the HP has been very positive. Although at times there is some complexity involved, the products have been easy to install, easy to use and functional. It should be pointed out that having access to personnel on IBM systems who were already experienced with LU6.2 was a great advantage in implementing the LU6.2 capability on the HP.

Some observations:

* Partnership

  As has been stated several times, both from a technical programming aspect and an applications viewpoint, program-to-program communications is a partnership game. Other approaches will not work.

* Teamwork

  It will seldom be the case in any organization that any one person will have all the IBM and HP applications, networking, operating system, operations and support skills necessary to single-handedly implement a successful HP/IBM APPC project. It will be necessary that people work together that may have little experience in each other's fields. A degree of patience, mutual respect and a commitment to success will be big helps as well as the recognition that it will take a little time for each to learn from the other.

* Operations

  There has been nothing said so far about operating the LU6.2 utility software. There are new subsystems on the HP3000 that the operator must become familiar with and learn how to start, stop and check. Part of the LU6.2 package is a standard set of UDCs designed to "plug in" at a location and be easily integrated into site procedures. In addition, there is training provided for the operators and system manager at a site to understand the SNA and LU6.2 environment.

* Installation

  In the same way that the software is standardized, installation and configuration are standard as well as much as possible. This means that all of the HP3000s are essentially defined the same way to the SNA network. This simplifies the HP configuration, and makes support and problem solving easier.

FUTURE

The LU6.2 utility will continue to be deployed to more applications and sites as business needs warrant. The prospect of being able in the future to have LU6.2 conversations with other SNA nodes and not just with the IBM mainframe, raises some possibilities that appear very interesting and may provide some opportunity for LU6.2 based program--to-program communication in an environment that does not include an IBM mainframe.

Remote Process/Job Scheduling using NETIPC

By Garry Osborne
Michael Mazza

Brown & Williamson Tobacco Corporation
2600 Weaver Road
Macon, Georgia 31298
912-749-8845
912-749-8682

Abstract

Distributing information is a challenging endeavor. Merely transferring files either manually or electronically only touches the fringes of integration. In order to facilitate the distribution of information and to integrate processes a method of scheduling and controlling the transfer process is required.

This paper will describe the business requirements for distributing information, and how we satisfied those requirements. Information is gathered on the factory floor via HP-UX systems and passed on to MPE systems for processing. Other information is downloaded from the MPE host to the HP-UX systems. The NETIPC client and server programs detailed in this paper manage the job/process scheduling, system security, and data integrity. These programs are instrumental in the integration of our factory because now we can transfer and process information automatically.

1. INTRODUCTION

1.1. The Organization

Brown & Williamson Tobacco is organized in the traditional manner (See Figure 1.a). The Macon Georgia plant is the main manufacturing location, and the Engineering department is located in Macon as well.

The Engineering department is responsible for all of the computer activity at the Macon facility. The manufacturing process is broken down into three major categories; Primary, Fabrication, and Operation Services (See Figure 1.b).

The computer network in Macon follows the distributed data processing philosophy (See Figure 1.c).

1.2. Network Configuration

1.2.1. Broadband

The Broadband installed in Macon is a standard 802. 4 LAN. The three major channels are defined as follows:

Ethernet Service at 10 Mega Bits utilizing three 6 Mhz Channels
Terminal Service at 5 Mega Bits utilizing one 5 Mhz Channel
Video Channel

1.2.2. 802.3 Baseband LAN Sub Networks

The Baseband LANs are connected to the Broadband via filtered bridges. Each Baseband LAN serves a single area or department at the Macon Branch. The Baseband LANs consist of segments of thin LAN coax cable which is flexible and easy to install. Each computer is connected to the Baseband LAN via a Lanic Card and Thin Mau. Some areas have fan out boxes or thin LAN repeater hubs which increase the flexibility and reliability of the Baseband LAN. The following is a list of the Macon Branch Baseband LANs:

MIS Data Center (six nodes)
Fabrication Computer Room (five nodes)
Fabrication Area (multiple nodes and terminal servers)
Primary Department Computer Room (four nodes)
Shipping Department (seven nodes)

1.2.3. Nodes

There are thirty computers on the Broadband LAN and several more terminal servers. The LAN serves two basic functions; one is providing ease of connection to any computer system from any terminal, and the other is the easy movement of data from computer to computer.

Currently there are only a few systems that require data from multiple computers. The main HP 3000 computer and the Fabrication HP 9000 computers are the most important.

2. APPLICATIONS

2.1. HP 3000 Systems

The HP 3000 computers support Macon Branch level applications. Most of the systems are traditional transaction oriented applications. Payroll, Personnel, Scheduling, Specifications, Inventory, and Quality are examples of the types of applications.

The Finished Goods Inventory system is one of the applications that utilize the NetIPC programs. There are many inputs into this system and one of the inputs is via NetIPC. The feeder application runs on a HP 9000 that is located in one of our warehouses and it supports that warehouse inventory system.

2.2. HP 9000 Systems

Most of the HP 9000 computers are department level systems. They provide stand alone processing capabilities for these departments. The HP 9000 HP-UX system combined with a relational data base provide the departments with the tools and processing power to work independently. This independence is an advantage in that backup schedules and maintenance can be coordinated locally. A disadvantage to this strategy is the lack of central control or a central repository for all Macon Branch data.

The Warehouse Inventory system is an example of a department level system. It runs on an HP 9000 Series 835 and has several terminals, printers, and radio frequency terminals attached to it. The system manages the receiving, storing, staging and shipping of finished goods located in the warehouse. Daily the warehouse inventory is uploaded and merged with other inventories to produce the Macon Branch Finished Goods inventory.

2.3. HP 1000 Systems

The HP 1000 computers provide both department level processing, and real-time capabilities. Some of these systems are localized to specific areas within departments and pass data up to department level HP 9000 systems. Other systems pass the data directly to the HP 3000.

The Primary Bulker Inventory system is one of the HP 1000 computers that use NetIPC to transfer data. It keeps track of the raw tobacco inventory that is being processed for Fabrication. Various field devices monitor and weigh the tobacco at set intervals. This data is then summarized, collected and transferred to the Macon Branch HP 3000 to feed the overall Primary Department Scheduling system.

3. BUSINESS REQUIREMENTS

From a business standpoint information or data is a valuable resource. It is even more valuable if it is shared or distributed.

Brown and Williamson Tobacco invests heavily in factory automation and real-time systems to help reduce direct costs. Most of these systems that are installed reduce labor costs or improve production efficiency which result in higher profits.

By linking these systems together and dispersing the information Brown and Williamson has been able to leverage their initial investment in automation. The NetIPC programs and processes like them provide us with the opportunity to begin plant wide integration.

The dream factory is one that is totally automated where information flows up and down through the computer systems as needed with little or no human intervention. The computers provide the data for making decisions to the appropriate level of management with plenty of time for management to respond.

There are several obstacles to overcome in order to achieve this dream. Ongoing production demands is probably the biggest obstacle. "What ever you do, don't stop production" is the battle cry of many General Managers. Our production managers are supportive of factory automation, but they are totally committed to producing cigarettes. Obsolete computer systems provide even more challenges. The state of the art systems that we installed two or three years ago are doing their job, however they provide little opportunity for integration. The ideal way to access this type of data would be with a data base engine, or through remote data base access. This is not possible with the current configuration of computers. This combined with the manufacturing process running 24 hours a day provided us with an interesting challenge. Upgrading these types of factory systems is equated to rebuilding an engine while it is running.

In summary our business demands that we stay competitive and automate, while at the same time keep producing cigarettes. One easy way to achieve this goal would be to build a brand new factory (which would probably be obsolete before it was completed) and avoid the retrofitting of computer systems. Another way to achieve the goal is to build bridges between the old and new systems. The NetIPC programs serve as bridges for our systems.

4. NETIPC SOFTWARE

4.1. HP 9000 Programs

dstoany	This function calls DSCOPY to any designated computer system. The program calling this function must have stdin redireceted to accept input from a text file. The text file has the copy descriptor for the DSCOPY command in the format described in the Networking Reference Manual.
hpnetjs	This function attempts to execute a job stream file on any of the specified HP 3000 computers.
nipcdaemon	This program is the NetIPC server which executes in background on all of the HP 9000 computers connected to the LAN. It creates a socket and names it "BWNETIPC", and then waits indefinitely for a connection request.

4.2. HP 3000 Programs

HPNETIPC	This program is the server, and it is executed in the background and runs continuously looking for work from the HP 9000.
HPNETJSP	This program is the client, and it called after a job stream DSCOPYs a file to a remote system. It then performs the function of submitting a job or process on a remote computer.

4.3. HP 1000 Program

HPNETCLN	This program is the client and performs two functions. First it does a programmatic DSCOPY call to transfer files to the HP 3000 and then it requests the job stream on the HP 3000 to process the data.

4.4. System Flow

HP 9000 to HP 3000

The HP 9000 application services the local system users and collects data. At specified intervals, or on demand the data is summarized and sent by the application to the HP 3000 using DSCOPY. After the file is successfully transferred, the application initiates the NetIPC Server on the HP 9000. The server submits a job on the HP 3000 via the NetIPC Client program that is running in a monitor mode on the HP 3000 computer. The NetIPC Client program on the HP 3000 is a job that is automatically submitted every time the HP 3000 system is initialized. It runs in a background mode waiting for work from the remote systems.

HP 3000 to HP 9000

The HP 3000 application serves the local system users, and at specified intervals downloads information to the department level computers. The information is usually extracted from Macon Branch level data bases, and occasionally from Corporate data bases. It is product specification type data that is common throughout the company.

The process on the HP 3000 is usually started by streaming a batch job. The job usually extracts data from an Image data base and then calls DSCOPY to transfer the file to the remote system. After the file has been sent the application sends a request to the HP 9000 client to start a process.

HP 1000 to HP 3000

The HP 1000 usually performs some real-time functions and the application serves the local system users. At scheduled intervals, or on request the data is summarized and sent by the application to the HP 3000 using a DSCOPY call. After the file is successfully transferred, the application initiates the NetIPC Server on the HP 1000. The server submits a job on the HP 3000 via the NetIPC Client program that is running in a monitor mode on the HP 3000 computer. The NetIPC Client program on the HP 3000 is a job that is automatically submitted every time the HP 3000 system is initialized. It runs in a background mode waiting for work from the remote systems.

4.5. Security Issues

Allowing remote computers to schedule jobs or processes creates possible security issues. The NetIPC client programs in place at Brown And Williamson Tobacco allow remote processing on both HP 3000 and HP 9000 computers. The HP 1000 computers currently send data to the HP 3000 only. The appropriate NetIPC client programs are in place for these systems, however there is no requirement to pass data in that direction.

HP 3000 Security Issues

The security on the HP 3000 is handled at the user level with the help of VESOFT's Security 3000. The NetIPC client job is submitted every time the HP 3000 is booted and it runs in background until it gets a request from one of the remote system servers. The job is run under a special user that has limited capabilities through our implementation of Security 3000. The only thing the user can do is stream jobs that reside in it's home group. In addition the jobs must be set up within Security 3000 as valid jobs for this user.

HP 9000 Security Issues

The security on the HP 9000 is handled at the user level with the standard UNIX security. The NetIPC client job is executed every time the HP 9000 is booted and it runs in background until it gets a request from one of the remote system servers. The NetIPC client program has a specific directory that only contains valid programs that it can execute when called for by a remote server. The directory is maintained by the system administrator and it requires root user capabilities. This is somewhat limiting because every time a new process needs to be added the system administrator must add the program. A planned enhancement is to add a small data base of valid jobs and the related programming to maintain the data base.

The LAN is somewhat secure because the only computers or terminals attached to it are owned by Brown and Williamson Tobacco. We are not part of a Wide Area Network and we do not allow outside agencies to connect to our LAN. Each computer has its own level of security as does each application. In summary we feel fairly secure and our internal auditors are satisfied that we are adequately protecting the companies data.

5. TESTING AND INSTALLATION

The testing required test data bases and simplified NetIPC programs to simulate the movement and processing of data. We used the simplest application as the pilot and once it was accepted it served as a model for the remaining applications.

It took us several weeks to debug the client program on the HP 3000 and we felt as though we were pioneering the process. The documentation was nil and support from our Hewlett Packard Account System Engineer was limited because of his lack of experience. However the cooperation level was excellent and everyone was committed to completing the project. It was a critical task at one point because we were in the process of upgrading our HP 3000 from a Series 70 to a Series 955. The critical factor was the replacement of some old PTOP DS programs with the NetIPC programs because PTOP is not supported on the MPE-XL computers.

Installation was fairly smooth because of the extensive testing and we monitored the process for several shifts without any problems. The most critical application was installed in full parallel mode for one week and the users verified the data each shift. Once they were satisfied that the data was accurate and reliable we were able to discontinue the manual entry. Probably the most critical factor in the success of the installation was the commitment of our users. They saw the benefits to the automatic movement of data and were as committed as we were to make the project a success.

6. NETIPC IN A MANUFACTURING ENVIRONMENT

The manufacturing environment requires high system uptime and high system availability. Some of our systems run 24 hours per day 7 days per week. The manufacturing areas depend on current data to do their jobs. The NetIPC data transfer and remote job scheduling subsystem supports this environment. After the applications are linked and the testing is complete it becomes an invisible process. The process was written generically so that additional applications from various computers could be added easily.

After the systems are linked and the process is seamless the challenge is how to keep all of the systems on-line and active. The network becomes the system or the resource required to process an application, not just one single computer. Backup times and preventive maintenance schedules need to be coordinated. Having reliable and or redundant equipment becomes a requirement. Overall the integration has many more advantageous, but it makes the system and LAN administration more complex.

7. CONCLUSIONS

The NetIPC system that was installed is a success. It allowed us to migrate from a HP 3000 Series 70 to a Series 955 while maintaining the computer to computer communication. It in fact was a vast improvement over the older PTOP programs and a lot more flexible. We have added one additional HP 9000 to HP 3000 transfer since the original systems were converted and it was easy enough.

The factory of the future is still a ways off for us, mainly because we are concerned with supporting the factory of today. NetIPC has provided benefits to us by allowing us to migrate one step closer to our goal.

Figure 1.a

BROWN & WILLIAMSON TOBACCO CORPORATION

- ENGINEERING
- PERSONNEL
- MANUFACTURING
- R&D
- FINANCE
- SALES
- MIS

Figure 1.b

MANUFACTURING CATEGORIES

<u>PRIMARY</u>

- .Leaf Storage & Processing
- .Casing, Cutting, & Drying
- .Flavoring & Blending
- .Cut Tobacco Storage

<u>FABRICATION</u>

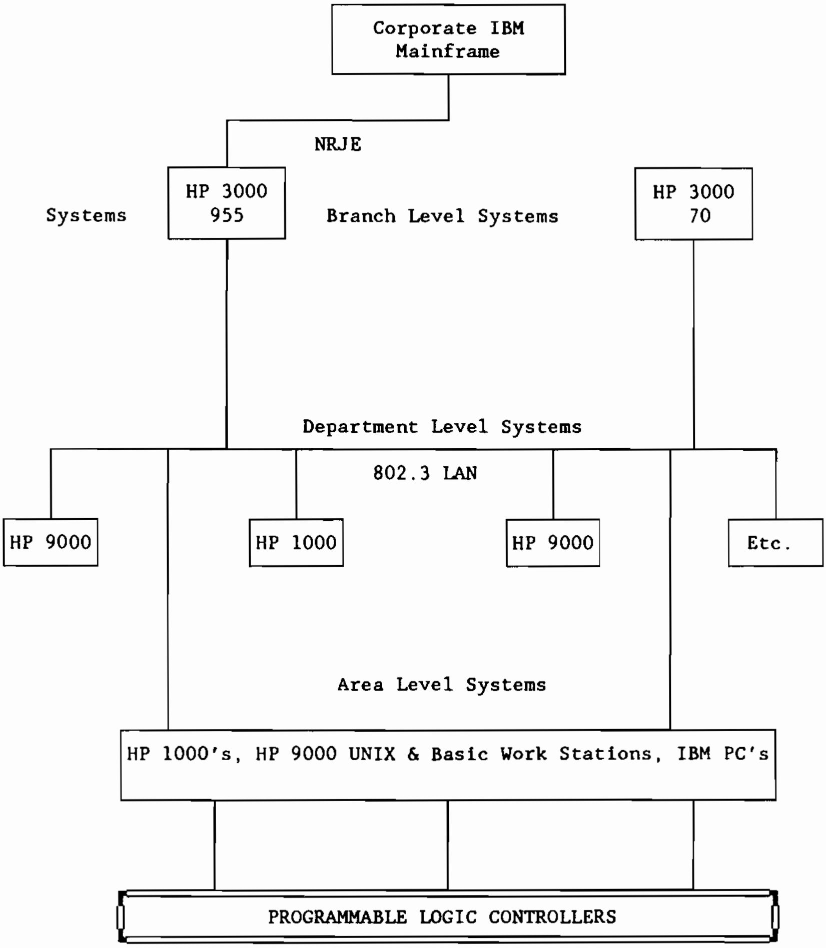
- .Production Scheduling
- .Product Specification
- .Cigarette Manufacturing
- .Quality Monitoring
- .Efficiency Accounting

<u>OPERATION SERVICES</u>

- .Purchasing Raw Materials
- .Manufacturing Material Inventory
- .Support Fabrication
- .Store and Distribute Finished Product

Figure 1.c

LEVELS OF AUTOMATION



TITLE: Survival Techniques For a 'No-Fault' HP 3000

AUTHOR: J.G. Page

Union Camp Corp.

Box 178

Franklin, VA 23851

804-569-4710

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 7117

X.25 - The Federal-Mogul Story

Tim "Ollie" Cahoon
Federal-Mogul Corporation
PO Box 1966
Detroit, MI 48235
(313) 354-7700

Federal-Mogul Corporation is a global manufacturer and distributor of precision parts. The principal products sold by the Company and its subsidiaries are grouped into two industry segments: vehicular and industrial components, and fastening systems.

Vehicular and industrial components include ball and roller bearings, engine and transmission products (engine bearings, pistons, bushings and washers), sealing devices (oil seals, and other high performance specialty seals), hot forged powdered metal parts, automotive fuel pumps and related systems, lighting and electrical components, and air bearing spindles. Fastening systems include fasteners, installation tools and power sources.

These products are sold to five major segments: aftermarket, auto and light truck; general industrial; heavy truck, farm and construction; and aerospace.

In 1985 a decision was made to decentralize the computing environments at the plants manufacturing our core products. These plants produce our engine bearings, sealing devices, and ball bearings. The bearing and seal plants did their processing using IBM 8100 workstations connected to the corporate IBM mainframe. The engine bearing plants which were using Burroughs computers even did much of their processing on the corporate system. Federal-Mogul was very much an IBM based company. The project team reviewed many different computer systems with the HP3000 being picked as the platform to use. During the next three years the new machines were installed, applications were converted, and new more complex systems were implemented. In a short time, a plant which did 90% of its processing on the mainframe, now only uses it to pass data to other divisions and to use the EDI system.

As the plants were installing their systems, an HP3000 was also being installed at headquarters. Just as with the plants, applications here were also being converted from the mainframe and new systems were developed. While data from the plants still remained on the mainframe it was easy to download it to support the new applications, but as the conversion efforts proceeded, file transfers from the plants to corporate HP3000 became more common. Since all the HP3000s were connected to the mainframe using SNA NRJE/3000, we used that facility to pass files, but limitations were soon reached in the types of data, and the size of the records that could be transferred. The headquarters DP staff also could not access the remote machines to assist in application development or troubleshooting systems. Engineering and Sales users were asking for access to the databases at the plants. All of these requirements led to the search for a way to communicate directly between the HP3000s.

The search for the perfect network began with a review of what the company was doing, and what it wanted to do in the future. Our large SNA network was working just fine, but SNA is an IBM not an industry standard. Future plans called for Unix workstations and personal computers located on local networks, to talk to other networks across the country and the world. The search ended with the selection of X.25 as the type of network to install. It provides redundancy, reliability, and is an industry standard. The use of additional software or hardware products would also allow us to run our SNA and Lan data across the network with no conversion, yet still retain the benefits of using X.25.

Once we made the decision to go X.25 we had to pick a network vendor. The process started with Hewlett Packard, Tymnet, and Telenet. Tymnet and Telenet are the two largest public X.25 networks in the United States and they connect to many countries in the world.

In setting up an X.25 network you first have to decide if you want it to be public, private, or a hybrid. Each choice has its own pros and cons. Going public means that you connect your computers to a public network such as Tymnet or Telenet. Your data passes through networks used by thousands of customers each day. This type of network connection reduces your cost for hardware and support since it is being handled by the vendor. You pay the network a fee based on connect time and characters transmitted. A private network is just the opposite. You purchase all the hardware, all the phone lines, and provide all the support. While you don't have to pay for connect times or characters transmitted you do have the hardware and software support costs. Hewlett Packard has a program that addresses the support issue by having their network support staff monitor your network. A hybrid network is one that is basically private in nature but has connections to and is monitored for you by the public network vendor. With a hybrid network you are passing your data over your own lines but allowing a network vendor to monitor and control it for you. This type of network also allows you to use the connection to the outside world for communications with users on the public network.

The review process took several months and at that time we selected Tymnet as our X.25 vendor. We liked their offering to create a hybrid network and we also liked their equipment. Tymnets hardware is software driven and by just downloading the correct software we can connect X.25, SNA, and asynch devices without having to make a site visit.

The conversation now became how to implement the new network. It was decided that it would be preferable to do it with out disturbing our current SNA network, yet be done so that once X.25 was in place we could roll SNA into it. It was decided to use our current star network configuration, the IBM mainframe as the hub and SNA lines going out to each plant, as a place to start. We already had dedicated phones lines to each of the plants for our SNA link and with some investigating found that using CODEX modems could make our 9600 baud lines run at 19200 baud. A feature of the modems allowed us to split the bandwidth of the line in up to 4 channels. We could run the SNA on one channel and leave it connected to the IBM and connect the X.25 channel to new network. This would meet the requirement of letting us run our SNA network undisturbed and yet still add X.25 using the same phone lines. Later on, once the new network is stabilized, we can add SNA over X.25, unsplit the line and maximize its usage. These particular modems are also monitored and configured using a PC based system. This gives our network control staff the ability to see hardware and phone lines problems from a central site.

Other things were also happening during this time. Tymnet was busy installing our equipment, configuring the network and establishing the correct security parameters for our new X.25 network addresses. Security is not taken lightly by any of the public networks. Hewlett Packard was also busy setting up their network implementation plan to make sure their end of the project was successful. I learned that NS3000 uses an HP proprietary version of TCP/IP and that protocol requires each host to have a unique IP address. TCP/IP is commonly seen in the Unix based workstation networks used by universities, engineers, and scientists. The largest TCP/IP based network, and possibly the biggest network in the world, is called INTERNET, previously known as ARPANET. This network was originally developed for the military and is used by universities, government, and corporations to communicate research work to other members of the scientific community. If we were a small network without much chance of connecting to something like INTERNET, we could have made up an IP address to use. We also could have ordered IP addresses directly from Hewlett Packard. They have a bank of numbers assigned to them that they can distribute to customers. However since we felt strongly that we would be connecting to networks like INTERNET, or our customers private networks, it would be in our best interest to get the addresses from the Network Information Center at SRI International. This guarantees that the addresses would be unique and registered in Federal-Moguls name. SRI International is the organization that oversees INTERNET and is the keeper of the official IP address list.

With all that going on it was time to start working on the HP3000 configurations. I decided to make things as easy as possible since there were 10 sites to install and support. The first thing I did was to get a copy of each systems configuration using the SYSINFO program in TELESUP. This provided me with all the information I needed to assure an easy and uniform configuration across the machines.

One decision I made was not to lock the plant into using any special DRT number for the hardware. It would be good enough to standardize the LDEV numbers and give the plants the ability to manage their hardware without putting undue restrictions on them.

After reviewing the configuration listings I developed a convention to use LDEV 300 as the X.25 INP board, LDEVs 310-349 as VTERMS (host to host pseudo-terminals), and LDEVs 350-399 as PAD terminals or printers. I then copied the configuration dialog pages out of the NS X.25 Link Manual and modified them to meet my conventions. Only the options to be used were left in so that a person following it would have no choices to make. I then used those sheets to configure my machine and after making some corrections, I had them copied and placed into binders to be distributed to the plants. The test on my machine proved helpful in finding problems with the documentation and the sheets eased the update process for the plants. The documentation also assisted the HP CE installing the INP Board in each machine. Each one was able to install the hardware exactly as we wanted it, and it kept HP3000 configuration uniform between all plants.

The software was done in a similar manner. I worked with our HP SE to pick the best version of MPE that would meet our needs. We selected MPE V delta 4 since provided inbound PAD support. This feature allows someone in the public network to call in using a dumb terminal or PC and use our machines, barring security requirements. Once the operating system was selected, we ordered our Power Patch tape and other selected patches. Using the SYSINFO listing gathered previously, system tables that needed to be changed or reviewed were identified and the major tables for all the plants were placed in a spreadsheet type document.This document told everyone what tables to modify based on the recommendations found in the NS X.25 Link manual, and let the DP staff at the plant compare their system table settings to the other sites. It answered the age old question of; "Are my system tables set correctly?". The plants could compare their table settings with everyone else and make changes if they wanted. This worked out quite well since all the plants are running the same applications. Using the binder created for the hardware installation, I added in copies of the MPE installation instructions, Power Patch installation instructions, and our Patch tape installation instructions. I ordered FOS and SUBSYS tapes for each location and then duplicated the PowerPatch and regular patch tapes using a program call STORCOPY located in Telesup account. This program reads a MPE Store tape and copies it into one big disk file. You can then run the program in reverse and make multiple tapes from that one file. This process assured us that everyone would be updating from the same patches and that if any problem did occur, we could apply the solution across all the machines.

I convinced one lucky site to be the first to update to the new operating system. When the update was completed and the new patches needed were received, everything was sent out to the plants to be installed along with an implementation timetable. Each plant was to have the MPE update done, the INP installed, and the pseudo-terminals configured by their implementation date. The computer had to be down several times. Once to install the new version MPE, once to install the INP and configure it, and then one more time to finish configuring all those pseudo-terminals. They couldn't install and configure the INP earlier because they needed the new MPE device driver that comes with the product.

When configuring the VTERM and PAD terminals you should realize that in days past Hewlett Packard had you use the

LDEV number of the INP or Lanic card as the DRT number of these pseudo-terminals. Today you use the LDEV number of the system console for that DRT number. These pseudo-terminals are now the same for the LAN and X.25 software. On one system with the LAN software, we deleted all the old terminal definitions and used the new ones. Also some contributed software when run, especially older versions of SYSINFO, give you informative error messages that these devices have invalid DRT numbers.

One fun activity did result from our new software. NS3000 requires each machine to have a name. I decided to let each plant pick their own, at least within reason. Some were just picked, others had small contests, but all in all we had some fun with it. Our names are; Fred, Minnie, Isaac, Hokie, Wilbur, Scarlett, Sterling, Thomas, Augusta, and Mint.

As the plants were getting their machines in order, I was also busy getting the X.25 software itself configured. The ease of installing the software and hardware made me think that this would be a snap. The process configuring the network is definitely the most difficult. After going through it I would recommend to anyone just starting in networking to take the NS3000 classes. Just getting a handle on the terminology is worth the expense.

The NS3000 software uses files in the NET group of the SYS account. The configuration file is called NSCONF and is accessed using a program called NMMGR.PUB.SYS. This file has the same format as NMCONFIG.PUB.SYS which is used by the IBM SNA software. The NMCONFIG file also controls the network logging parameters for all the datacomm products. Working with our TYMNET representative we kept our X.25 configuration parameters at the default and/or what HP uses as defaults in their screens. The manuals were little help in this endeavor as they fail to give any comprehensive list of the X.25 parameters you could change, what screens to find them on, and how the software treats them. I finally got a copy of "X.25 - The PSN Connection" which proved to be a great X.25 primer and the best source on how HP implemented X.25. The only X.25 facilities we used was Throughput Negotiation. The X.25 parameters you set in NSCONF have to match those your X.25 vendor has configured for your network connection. It took several phone calls and listings to make sure we had it all set correctly. NSCONF also contains all the X.25 addresses of the hosts you wish to connect. These addresses are associated with a key value, usually a shortened form of the hosts name, that you also have to enter. Next you have to enter the complete host name along with its IP address into the Network Directory File called NSDIR.NET.SYS. Every host in your network that you wish to communicate with has to have its entry in the Network Directory and its corresponding X.25 address added to NSCONF. Once that is complete and you have plugged your INP cable in to the network you can begin to test the software. The first test will be to see if the correct Call Request packets are being sent from your machine. This can be done with either the DSLINE command with the SERVICES option or using the DSLINE command with a REMOTE HELLO.

Example:

```
        DSLINE MYHOST;SERVICES
 or
        DSLINE MYHOST
        REMOTE HELLO MYUSER,ONREMOTE.SYSTEM
```

This sort of testing helped get some of the initial bugs out of the system. Your vendor can watch the packets generated by your machine and see if everything is in the right positions with the correct values. If everything looks good then you probably have NSCONF configured correctly and it is now time to hit the road and connect the first plant to the network.

The first plant to be connected was selected because it was only a two hour drive away and easy to reach. I went to the site, created the NSCONF.NET.SYS file using NMMGR and filled in the values by using the NMMGR Guided Tour function. Since everything was kept at the default values only addresses and the use of the THROUGHPUT NEGOTIATION facility had to be entered into NSCONF and NSDIR. I only entered in the address of the headquarters host into the configuration files at this time. Later on after the software was working I would use another function of NMMGR to download the complete address list from the configuration files on the corporate HP3000 into the configuration files on the remote machine. Next we plugged the INP cable into the modem and with Tymnet watching the packets on the line we brought up the NS3000 software and attempted our first remote logon. To my surprise it worked first time. This process was then repeated at each of the other sites with generally the same success.

When you have done several installations it is a good idea to review both the host and network configurations. You should now have a better idea of how the host and network software works and it should be easy to find mistakes and make corrections. We had a situation where the Tymnet configuration for the our first site had some limitations built into it due to uncertainty that the new modems would arrive in time for the installation. The computer at that location worked fine across the network but it was always slow. We discovered that the configuration specified a top throughput limit of 4800 baud rather than the 19200 baud we desired. Some of the hosts also had bad IP addresses that kept them from calling certain machines. These also had to be identified and corrected.

Our biggest problem currently is dealing with the differences between how NS3000 and SNA NRJE/3000 works. With the SNA network our users can stream a job on the HP that builds a file, submits an IBM job to transfer the file, and continues processing. When the IBM job is submitted to SNA the HP job can continue without problems getting the file up to the IBM is now the responsibility of the SNA software. With SNA if the link is down to the IBM it will hold the job until the link is brought back up so that the transfer can occur. The same job using the NS3000 DSCOPY program to pass the file requires different techniques. With NS3000 there is no other process to pass the transfer step off too as with SNA. When the transfer occurs, it happens right then and if the remote host is unavailable at that time you either have to accommodate that fact in your job stream or have it abort.

Security is also a big issue with our remote sites. Many are worried that people from the public network will hack into our machines through the public network connection. Others just don't want anyone on their machine without their knowledge. We have been working to solve those issues by tightening security on the machines and using the capabilities of Security/3000 from Vesoft to keep people out. Using Security/3000 features we have create users on each system with very basic capabilities that the DP staff can use on each machine. Security/3000 compares the user name with the machine they are logging on from and logging on to with its database. If the access is not allowed they are logged off and an entry recorded in the violation file. If the access is accepted the user is placed into the application.

Spooling is also another issue that causes problems within the network. While NS3000 has great facilities for passing files, doing remote logons, and accessing databases remotely, it suffers when you want to move spoolfiles between machines. When run you a program on one machine you can use a file equate to direct the spoolfileto to a remote system. The problem with this is that the program will abort if the network goes down, and the programs speed of execution could be severely restricted by how fast it can write the data across the network to the remote spoolfile. The second solution is to have the program write its output to disk, which can then be copied using DSCOPY. The third and best solution is to use a 3rd party product which will transfer spoolfiles directly from the spooler of one machine into the spooler of another. This solution allows programs to run at the speed of the machine on which they are executing and then takes care of the spoolfile transfer automatically for you.

Other products on the market provide enhanced capabilities or cost savings. If you are only going to do remote file access you should look for products such as Quest Softwares NETBASE software. This product increases the file I/O through the network significantly. Spooling problems could be solved by using Carolians OMNISPOOLER and INFONET packages. You could also use SPOOLMATE from Unsion. Many job scheduling packages also have networking capabilities built in as well. If you need to know when a remote system goes down CALLBACK/3000 by Design/3000 is available. Outbound pad service while not provided for by Hewlett Packard can be found by using a package called PADEMU by Hermes. This package is distributed by The Solution Centers. If all this software and hardware seems to much to your daily file transfers, check out the NETWORK ENGINE by Telemon. As you can see there are many products available to enhance the functionality of your network. Use of these packages instead of the NS3000 software could give you real dollar and performance savings.

After fighting the battle once, I have summed up what I consider some key points for those who still feel the need to press on.

1) Take the NS3000 classes from Hewlett Packard. The configuration and design of networks requires a lot of work and you need to learn the terminology. If you are just hooking up one or two machines, it is worth paying HP to do this for you. If you are installing a large network you will need all the training you can get. Networking is a lot more difficult than you first perceive it to be. While you can read the manuals and learn a lot from the software, network design and configuration requires information outside the scope of those manuals.

2) You need to take some X.25 training to also learn the X.25 terminology. You will be working with your X.25 vendor

closely and many times have to figure out how and where the information he requests is presented on the HP3000 or in its manuals..

3) Take a TCP/IP class. TCP/IP is a protocol at a higher level than X.25 that makes sure your data gets to the right location correctly. NS3000 uses an HP proprietary version that doesn't talk to anyone else. There are thousands of TCP/IP users, but if you want to talk to them from a Classic HP3000 you must buy the WIN/3000 software from Wollongong. HP is suppose to create its own version for the Spectrum series in late 1990. The TCP/IP terminology is used throughout the documentation and configuration process. It really helps to understand what the manual is talking about. A company called Advanced Computing Environments has a two day class entitled, Introduction to TCP/IP. You can reach them in Mountain View, CA at (415) 941-3399. (Recent literature received indicates that Advanced Computing Environments may have changed its name to INTEROP, Inc. INTEROP is also the name of a large networking conference sponsored by ACE.).

4) Order a manual from HP called: "X.25: The PSN Connection" part number 5958-3402. This manual is a great primer on how X.25 works and gives you everything you need to know about using it. IT IS ALSO THE BIBLE ON HOW HP HAS IMPLEMENTED X.25 IN THE NS X.25 LINK. It has more information on X.25 than any of the NS manuals you get with the software. You should also try to get your hands on APPLICATION NOTE #425, an X.25 application note written for the HP 4951C and HP 4952A protocol analyzers, entitled "The Care and Feeding of Your X.25 Network". This document was written by the HP division that produces this equipment. This document gives an excellent description on how to troubleshoot a X.25 packet network and should be added into the NS X.25 Link manual. I discovered this application note information while attending a San Francisco Interex Conference. You should read paper number 4660, "Procedure for Gaining Control of an X.25 Network", by Kimberly D. Weinmann-Davidson of Hewlett-Packard Colorado Telecommunications Division. While the paper can be found in the conference proceedings, you may also want to get the audio tape recording of the session as it was presented at the conference. It was during this session that she handed out this application note regarding the troubleshooting of networks. I have used the information many times to not only solve problems but to assist in connecting new devices to the network. Its step by step approach is something you really need to solve problems quickly.

5) One last thing to remember about networking products. When you buy something like NS3000 and NS X.25 Link, Hewlett Packards responsibility is to get the hardware and software installed. They do not configure the network for you! You are responsible for network design and configuration. HP has developed support products you can purchase to assist you if desired but don't expect your SE to do the network configuration for you unless you purchase one of these products. I have to say our SE was helpful when we had problems and did the best that he could but he couldn't get too deeply involved.

I hope this paper has been some help to you. Below is a partial list of vendors mentioned in the above paper. This list is not meant to be complete but a starting point to help you in contacting a variety of people with products that could benefit you.

Vendor	Product or Service
Network Information Center SRI International 333 Ravenswood Ave. Menlo Park, Ca 94205 1-800-235-3155 1-415-859-3695	TCP/IP Addresses
Quest Software 640 Newport Drive, Suite 890 Newport Beach, CA 92660 1-714-720-1434	NETBASE software
Carolian American Drive #5 Mississauga, Ontario Canada L4V 1T8 1-416-673-0400	Omnispooler & Infonet
UNISON 675 Almanor Avenue Sunnyvale, CA 94086 1-408-245-3000	Maestro & Spoolmate
Telemon 492 Ninth Street, Suite 310 Oakland, CA 94607-4098 1-800-622-0630	Network Engine
Solution Centers, Int'l PO Box 160488	PADEMU HP pad software

Sacramento, CA 95816
1-916-622-0630

Design/3000, Inc. 1214 Hawthorne Ave. N.E. PO Box 13086 Salem, OR 97309-1086 1-503-585-0512	Callback/3000.

Title: File Servers in a Client/Server Computing Environment

Author:

Gene Jung
Hewlett Packard
700 71st Avenue
Greeley, CO 80634
(303) 350 - 4000

Abstract:

The Client/Server environment of the 1990's conceptualizes Clients consuming or accessing resources delivered over a network from Servers. With the increasing computing power of PC's and UNIX workstations, their data generating capacity increases also. By linkage in a network, these growing islands of information are joined to share resources and data. In the Client/ Server environment, file servers which only share disks should evolve to be an easily administered central repository of information.

New storage technologies are becoming available that can store many gigabytes of data in a cost effective form. Optical disk drives and optical disk libraries can provides gigabytes of disk storage. By utilizing these technologies in a centralized file server, administration of the data can be made much easier and managed more rigorously. With these technologies available, implementing a hierarchical file management system becomes actually feasible with a direct access secondary storage medium. In enabling this amount of data to be online and under control of a computing system, availability of data that was previously relegated to tape libraries can now be brought online and maintained in a managed system.

File Servers in a Client/Server Computing Environment

During the 1990's, the computer industry will experience another computing environment that is called Client/Server computing. During the 1960's, host/mainframe based computing became the predominant form for information processing. During the 1970's, technology created the minicomputer which enabled departmental or distributed computing. As customers took advantage of this technology, growth of host/mainframe computing declined . In the 1980's, technology created significant computing power at affordable prices for individuals with the introduction of PC's and workstations. This continual shift in the price/performance of computing brought new opportunities and challenges and created an increasingly competitive market for vendors. During each period, some new technology developed and creative people seized upon the moment. Each era came at the expense of growth in the previous computing environments. At the beginning of the 1990's, the technology that enables another new shift in computing is the emergence of commonly defined networks. These enable a Client/Server environment in which islands of computing are coming together where the Clients share the resources and power of the Server over the network.

This shift into Client/Server computing is driven by networking technologies that have been standardizing the connections and the communication protocols. Networks are used for sharing limited and expensive resources such as peripherals and data. As each era of computing has brought processing power closer to the individual users, the need for auxiliary services and peripherals has also multiplied. The needs for data management and input and output are no different for the workstation user than the mainframe user. This need to share data, services, and peripherals has driven users to network together. It is these users who have built Servers out of retrofitted CPU's and so started toward a Client/Server computing environment. As analysts conceptualize this environment, the Client is a computing entity on a network that accesses a resource or service from the appropriate Server over a network. This paper will examine some of the opportunities and challenges for networked mass storage in this environment.

The Client

The prevalent Client is either the PC or the technical workstation in a network. The most prevalent Server is a retrofitted PC or workstation with additional disks drives used as a file server. While the installed base of PC's is much larger, the storage needs of the workstation user for storage are more critical as the amount of information generated by these knowledge users is proliferating at greater than 50% annually based on data from IDC. That data typically involves CAD designs, scientific simulations, and graphics based visualization of data rather than word processing, spreadsheets, and business databases which are the typical applications for PCs. This usage pattern by workstation clients fuels a need for large amounts of storage capacity. Because workstation clients are the largest users of storage in the Client/Server environment, they will be the focus of this paper.

The Server

The Server as a resource provider initially evolved from a need to share disk storage which is a relatively expensive resource that each client consumes at different rates. This need to share storage led to the practice of retrofitting a workstation with additional disks and using it as a file server. Often, this process starts as a workstation sharing duties as a file server. But as the file server began to be more utilized, the performance of the shared workstation deteriorated until it become necessary to dedicate it as a file server. This decision to dedicate the file server is largely based on the perception of acceptable performance by users of the file server on the network.

The utility of this centralized resource invited other applications for servers such as databases, e-mail, compilers, and printing. All these applications again deteriorated the perceived performance. When this point is reached, a more powerful server with larger and faster peripherals may be sought. Alternatively, another server may be purchased and dedicated for some specific purposes. By dedicating a server to a specific type of application, this server can also be tuned and optimized for that purpose. Today, there are upgraded workstations being used as general purpose servers as well as a few dedicated tuned servers for specific applications.

Retrofitted File Servers

The predominant use or application for file servers is still for disk storage. This is a preferred choice for data that needs to be shared or administered centrally. As workstation users continue to create more information, they have found that sharing large expensive disk storage devices over the network is still cost effective. As another benefit, this centralization makes the task of administering the data much more efficient than having data distributed widely over the network to clients and other servers. Administrative tasks such as backup, archiving, allocating space, and maintaining accounts for gigabytes of data can be significant tasks. Gigabytes of data generated by knowledge workers is a valuable investment that needs the same type of administration that mainframe and minicomputer users expect from their MIS staff.

A file server configured from a retrofitted workstation is the prevalent method of sharing storage over the network. While 2 to 25 MIPS of computing power is available to the workstation client, it is still viewed as such an expensive resource that dedicating a CPU for a specific server task must be carefully justified. Adding more applications and duties to this server may more fully utilize this resource, but compromises the performance of the server as more applications are added. At some point, this defeats the purpose of the server as an efficient resource server. Administrative tasks for this file server are usually left to a user who does not view the care of the file server as a primary job. The administrative tools available usually have been the normal software utilities distributed as part of the UNIX operating system. With a workstation that is not optimized as a server, administration is a secondary task with unfriendly software utilities. Therefore retrofitted file server is a solution that is only adequate up to a certain point.

Retrofitted File Server from Vendors

Recognizing that retrofitted file servers degrade to an unacceptable performance level due to greater utilization by more users or applications, some workstation vendors have offered more powerful general purpose file servers. These are bundled workstations with larger disks and high performance CPUs. As examples, Hewlett Packard has the 15NS and 25NS servers which bundle our Series 300 workstations with large disks and a backup tape drive. Hewlett Packard has also introduced the Series 635SV and the 645SV as file servers which bundle together our Series 800 workstations. These high performance file servers are popular for hosting several applications on the server. They are being used in ways that are similar to the services provided by a minicomputer. The Series 600's are most often used as database servers. Their administration is similar to that of a customer retrofitted file server. This type of system is flexible for different applications, but is not optimized for a particular one. The administrator must balance the usage of this server to make sure it offers acceptable performance and resources.

Specialized File Servers

Recently the Client/Server environment has seen the appearance of a few specialized file servers used as storage servers. These servers can be segmented into two categories.

The first is a high performance storage server. The high performance server is optimized for high I/O, both internally and over a network. It serves diskless workstations well as acting as their disk replacement over a network. It is able to serve many clients up to the point where the clients saturate the network enough to suffer a performance degradation. Because this type of server provides more clients with central storage, these servers are able to reduce the storage cost per seat. Centralization also makes the administration task easier and reduces data redundancy. But the declining cost of Winchester disks brings competitive pressure to this type of server. For frequently used applications and local data, the performance of a local Winchester disk is preferable if it is affordable.

The high capacity storage server with gigabytes of storage for the workstation client is the second specialized server on the market. The demand for this choice of storage server is fueled by the high rate of data generation by workstation applications. On the market now are optical disk libraries that utilize either WORM or rewritable optical disks. Both of these devices make tens of gigabytes of removable optical disk storage feasible at a lower cost per megabyte than Winchester disks.

To meet the constant demand for more storage, this new optical storage technology is being utilized in various applications. It is fundamentally different from direct access Winchester disk storage or offline tape storage. Rewritable optical disk libraries can store tens or hundreds of gigabytes in near-line storage. Data stored on these optical disk libraries can be accessed in seconds using the robotics of the autochanger. This is a great improvement over accessing tapes in tape vaults that must be retrieved by operators. Rewritable optical disk drives offer direct access storage with a removable, relatively inexpensive media that has a shelf life of at least 10 years. By using optical disk libraries, a few optical disk drives can access large numbers of optical disk cartridges in automated operations under a system's control.

The Needs

To reiterate, storage needs for workstation clients are growing rapidly due to their increasing data generating capacity as well as the volumes of workstations being sold. In addition, the technical workstation client has a need for storage that is high performance, expandable, reliable, cost effective, and capable of being accessed over workstation networks. As the number of workstations increase on networks, the transparency of file location becomes even more important. Finally, there is a need to effectively administer gigabytes of information on the typical workstation network.

The Opportunities

The emergence of standards in networking protocols and connections makes the choice of storage in the form of a high capacity storage server attractive. In the technical workstation market, the standard for distributed file system protocols is Sun Microsystem's Network File System or NFS and the ARPA services TCP/IP. Ethernet or IEEE 802.3

Lans are the standard network connections for workstations. By using these networking devices, it is possible to build a high capacity storage server for many workstation clients.

The availability of optical disk libraries with their gigabytes of storage becomes an attractive resource that can be shared if it is managed with imaginative software that takes advantage of the nature of a robotic system and hides the potential disadvantage of a relatively slow mechanical assembly.

Applications such as backup, data logging, archiving, and information libraries could take advantage of an automated robotic system that can load a large number of storage media either sequentially or randomly. Since there is the need to share the data, large capacity storage servers implementing these applications are being investigated today.

To hide the inherent relative slowness of the mechanical nature of optical disk libraries, hierarchical file algorithms can cache or store frequently used files to fast Winchester disks while keeping the bulk of the less frequently accessed files on the slower optical media in the optical disk library. This type of application takes great advantage of the random storage capability of the optical disk drive and utilizes optical disk libraries to their fullest potential. When implemented in this imaginative way, optical libraries could fulfill a great need for larger, cheaper online storage than has been commonly available before.

The Challenges

By combining the capacity of the optical disk library with the storage system of a Winchester disk based file server, a very large capacity storage server can be built. The challenge lies in integrating the capabilities of each component to create a whole that satisfactory addresses the needs of the workstation client.

Hewlett Packard is currently investigating such a combination. This storage server would manage files such that the most frequently used files accessed on the server are stored on the faster Winchester disks. The files that are infrequently accessed could then be stored in the slower optical disk library system. This system would lower the average cost of storage, centralize its administration, be expandable, and have the high performance for those files that needed it. With a large system of removable media and a potential for tens of gigabytes of data, utilities for automatic management of these files and media must be designed in. With access to some of these utilities, user defined administration duties can be easily automated or minimize operator intervention. Adoption of the technical workstation client's network standard means many clients can share this server's resources.

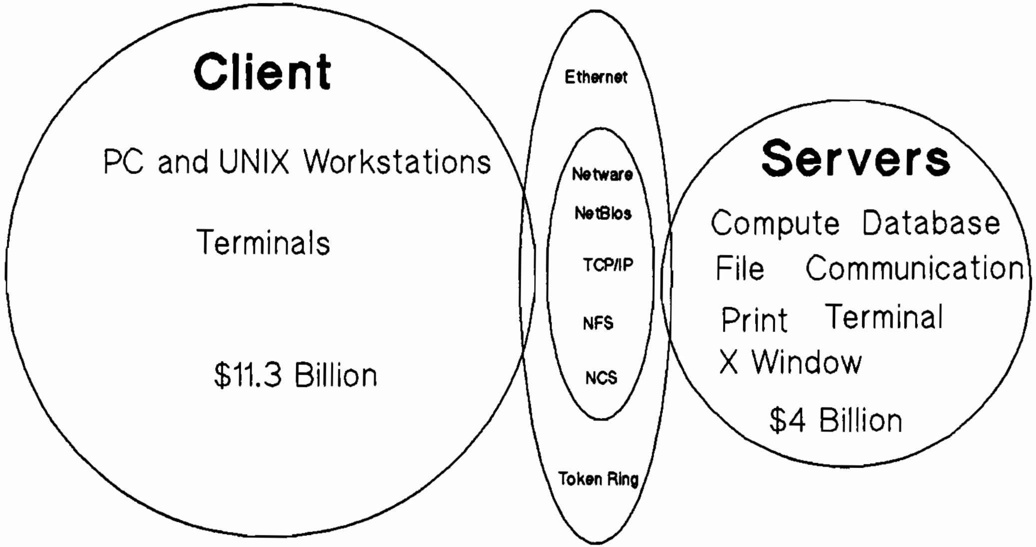
Summary

By using a large capacity storage server configured around the use of optical disk libraries, the storage needs of the workstation client can be served more efficiently than having many files stored on many clients' local disks or on many smaller servers.

- This much capacity means much more data can be shared.
- This type of expandable capacity on a central online location can bring much more of the data that was previously archived offline in tape vaults back online and available to all networked clients.

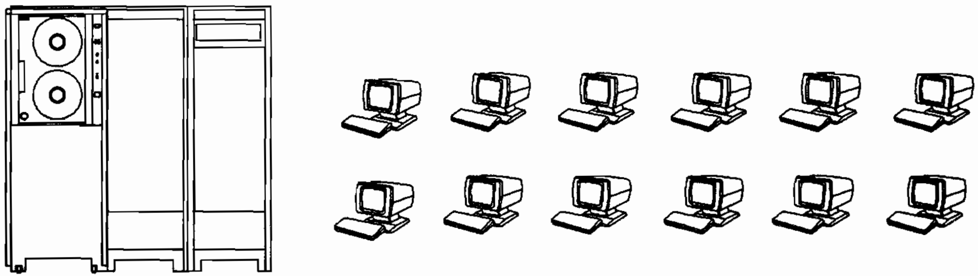
- Performance constraints can be avoided by maintaining very frequently used data on the client's local disk and moving the large data files to the storage server.
- Its administration can be flexible and automated enough to minimize or eliminate operator intervention.
- Clients can take advantage of the automated administration by moving their less frequently used files to optical storage on the server, but still have them online.

1993: Client/Server Market Revenues

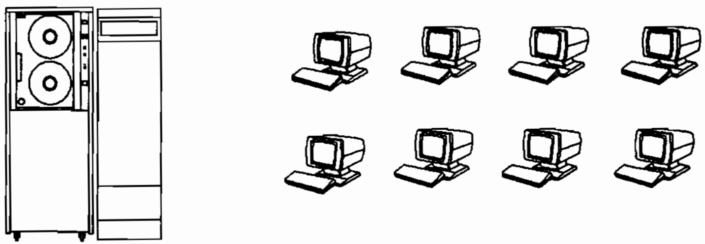


Source: Forrester's Professional Systms Report, October 1988.

1960's Mainframes



1970's Departmental Mini's



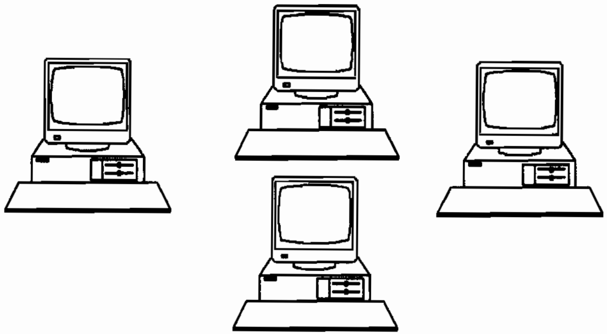
GREELEY STORAGE DIVISION

mainmini ej 4/90

HEWLETT PACKARD

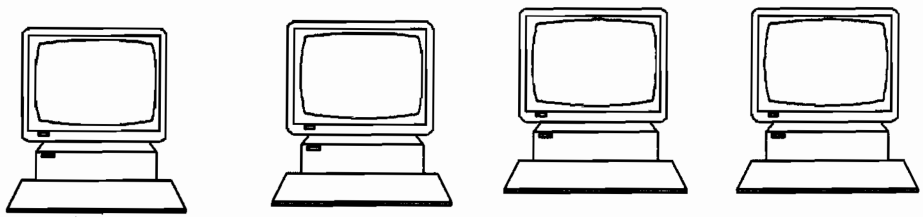
1980's PCs

.2 - 2 Mips



1980's Technical Workstations

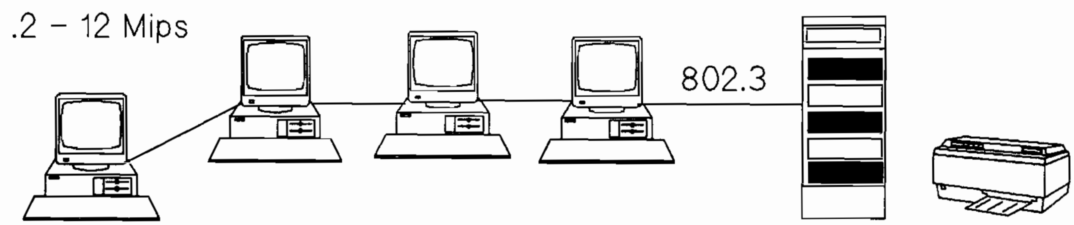
2 - 11 Mips



Computer Museum

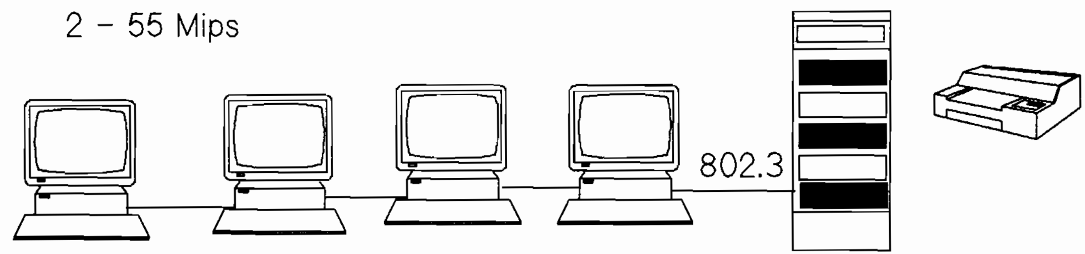
1990's Networked PC Clients

Servers



1990's Networked Technical Workstations Clients

Servers

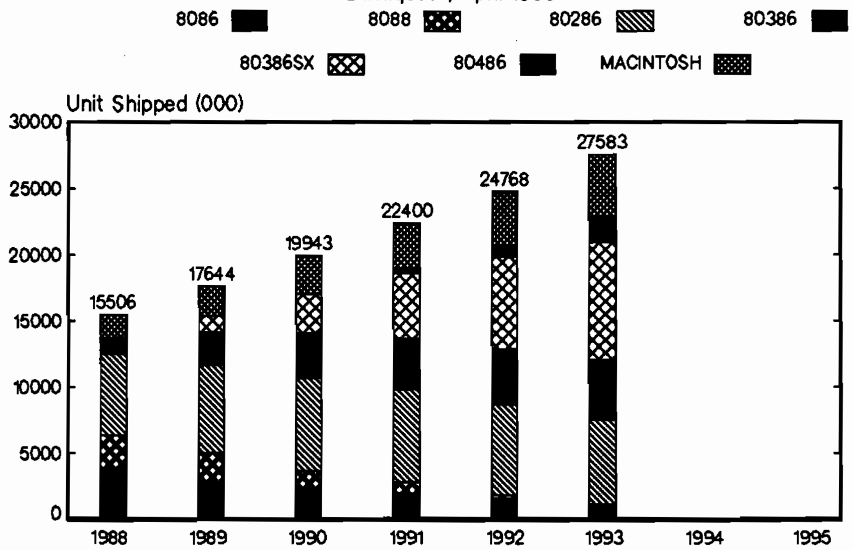


GREELEY STORAGE DIVISION
nettkpc ej 4/90

HEWLETT PACKARD

Worldwide PC Shipments by Processors

Dataquest, April 1989



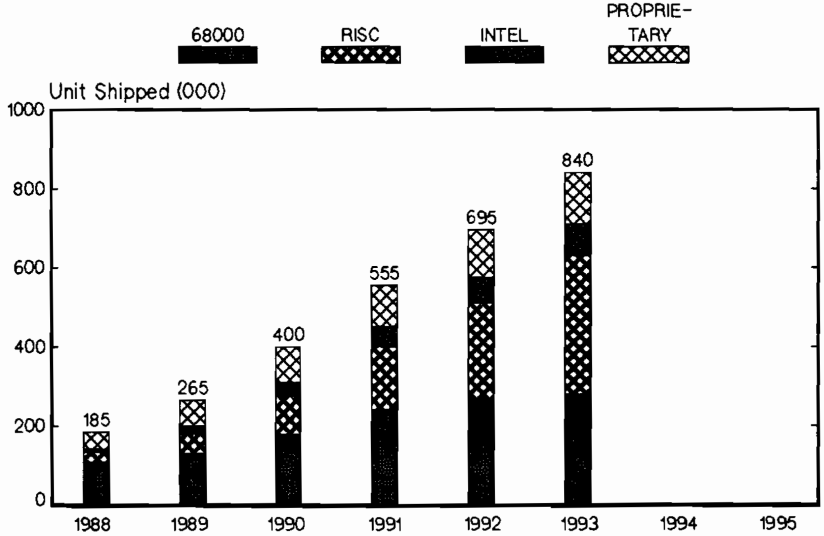
GREELEY STORAGE DIVISION

pcship EJ 4/90

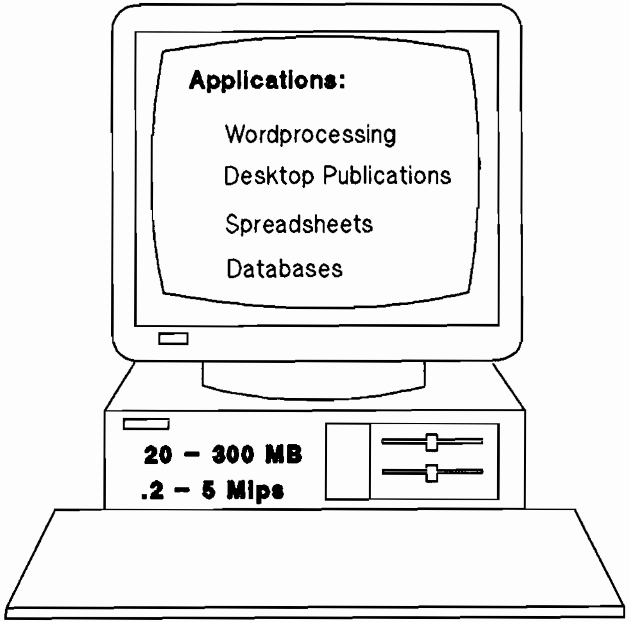
HEWLETT PACKARD

Worldwide Workstation Shipments by Processors

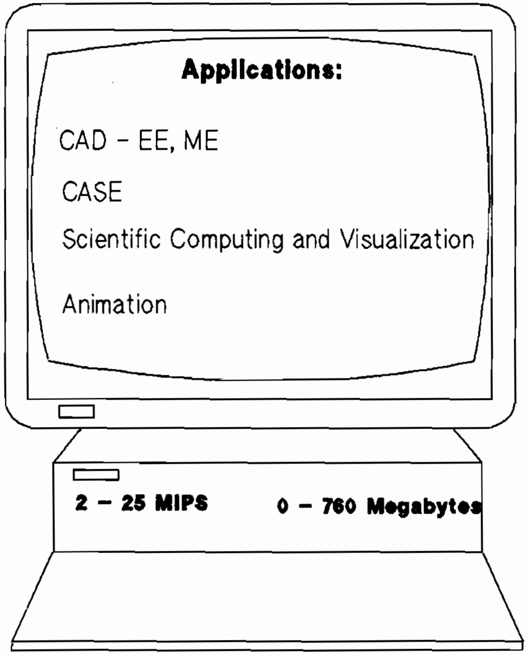
Source: IDC 1989



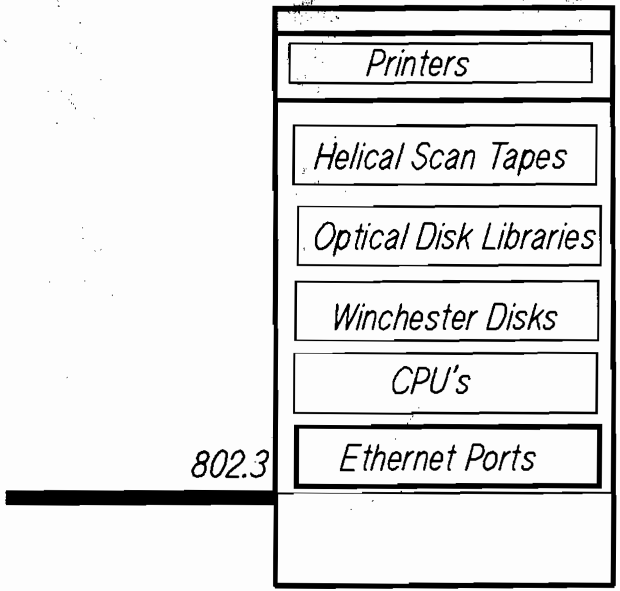
PC Client


Applications:
Wordprocessing
Desktop Publications
Spreadsheets
Databases
20 – 300 MB
.2 – 5 Mips

1990 Workstation Client

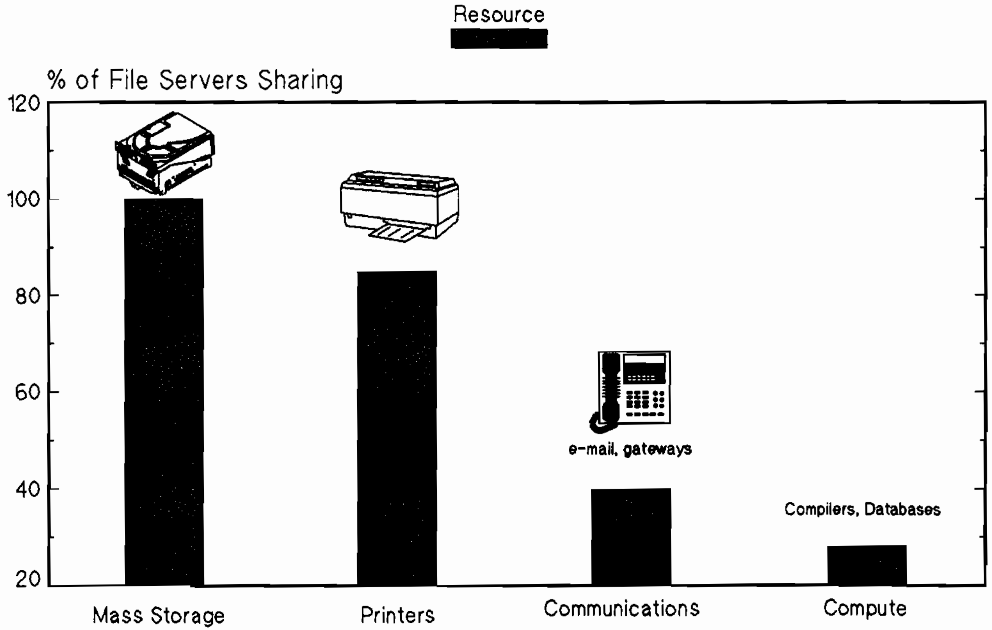


Resource Server

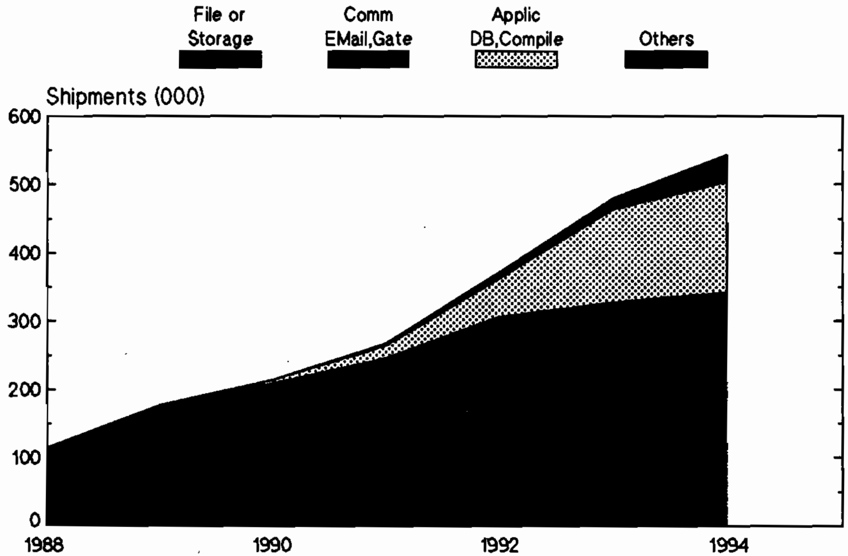


Shared Resources on the File Server

HP Marketing Studies from Uniforum



US Server Shipments by Shared Resources



Source: Forrester Research; May 1989

forsvr

GREELEY STORAGE DIVISION

FORSVR EJ 4/90

HEWLETT PACKARD

7143-17

Retrofitted File Servers

Built up from General Purpose Workstations

Satisfies Needs to:

- Share Expensive Peripheral
  - Disk, Printers, Plotters, etc.
- Share Data or Files in Workgroups
- Simplify Administration
  - Backup, Archiving, Accounting

Driven by Users

High Performance File Servers

Built by the Vendor Specifically
High Network I/O, High Disk Performance

Satisfies Needs to:

- Share Disks for
  Diskless Workstations, Large Workgroups
- Share Data or Files in Workgroups
- Simplify Administration by Centralization
  Backup, Archiving, Accounting

Competing with Declining Cost of Disks

High Capacity Storage Servers

A Technology enabled by the availability of:

- Standardized Network Connections and Protocols
  (802.3/Ethernet; NFS, ARPA Services)
- Low Cost Computational Platforms
- Fast Winchester Disks
- Affordable DASS technology
  - Direct Access Optical Libraries
    - Disk Access, but slower than Winchester
    - Large inexpensive capacity
- High Capacity Tape Devices (8mm, 4mm)
- Automated by Extensive Software Applications

High Capacity Storage Servers

Built by the Vendor Specifically

- High Data Storage Capacity

Satisfies Needs to:

- Share Disks for Large Data Generating Applications
  Images, Engineering Drawings, Simulations, Animation
- Share Large Amounts of Online Files
- Access Historical Data Online
- Simplify Administration by Centralization
  Backup, Archiving, Accounting

High Capacity Storage Servers

Enables Network Data Management by:

- Centralizing Data in a Single Location
- Keeping Very Frequently Used Applications and Files Local On the Client Disk
- Automating the Storage Of Archived Data
- Keeping Much More Data Online in a Managed System
- Utilizing the Optical Library Potential Now

The Storage Hierarchy

Traditional Mass Storage Segments

PRIMARY STORAGE

* HIGH PERFORMANCE HARD DISKS

- Access in milliseconds
- 5–10 dollars per Mbyte

SECONDARY STORAGE

* SEQUENTIAL TAPE DRIVES

- Access in minutes
- 1–15 cents per Mbyte

New Segment

DIRECT ACCESS SECONDARY STORAGE

* RANDOM ACCESS OPTICAL DRIVES/ LIBRARIES

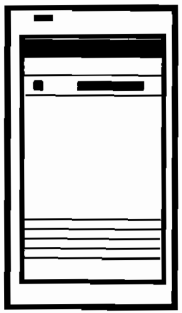
- Access in seconds
- 30–40 cents per Mbyte

Greeley Storage Division
STORHIER 1/90

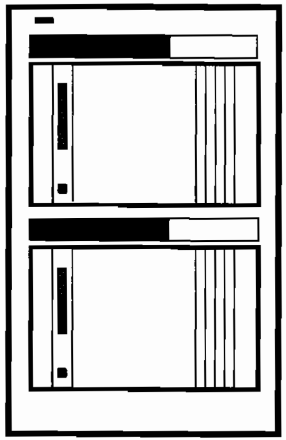
HEWLETT PACKARD

HP's DASS Optical Disk Libraries

20 Gigabytes



40 Gigabytes



Automated Fast Access

SCSI II commands

7 sec Swap

Lower Cost Online Storage

($2/MB)

Random Access

To each MO disk

On each MO disk

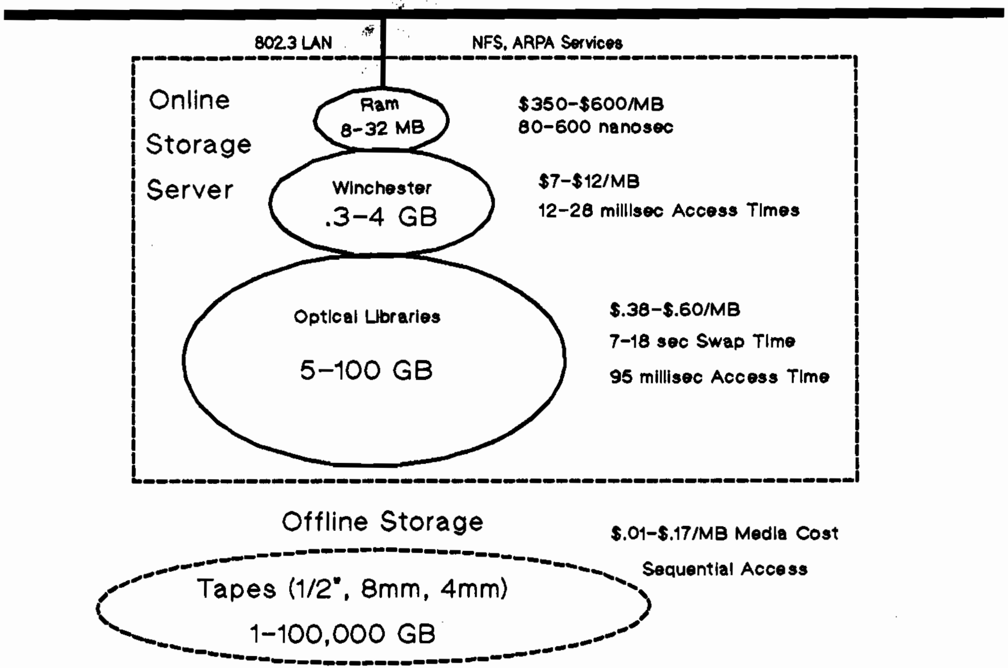
Modular

Stackable Libraries

Individual MO Disks

Hierarchical Storage Server

Enabled by DASS Optical Disk Libraries



GREELEY STORAGE DIVISION

HEWLETT PACKARD

Hierarchical Storage Server

Enabled by DASS Optical Disk Libraries

802.3 LAN

NFS, ARPA Services

Online Storage

[redacted]

Very frequently used files or records-sec,min,hourly

program segments, file segments

Frequently used files-hourly,daily

Current drawings,compiles,working files

Infrequently used files-biweekly,weekly,monthly,year

Images, documents, engineering drawings,

libraries, archived codes,

Offline Storage

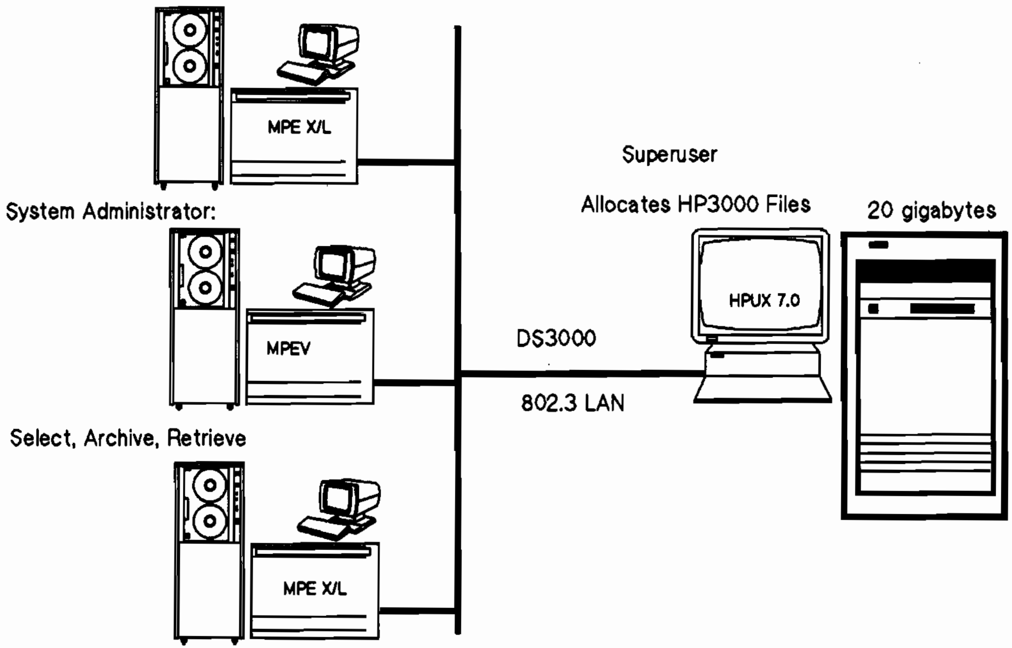
[redacted]

Very infrequently used files; system,file recovery

Backups, archived systems, copies

HP 3000 Archival Storage Server

Enabled by DASS Optical Disk Libraries



HP's LaserARC Server

Enabled by DASS Optical Disk Libraries

HP3000 Archival Storage

Infrequently used Databases

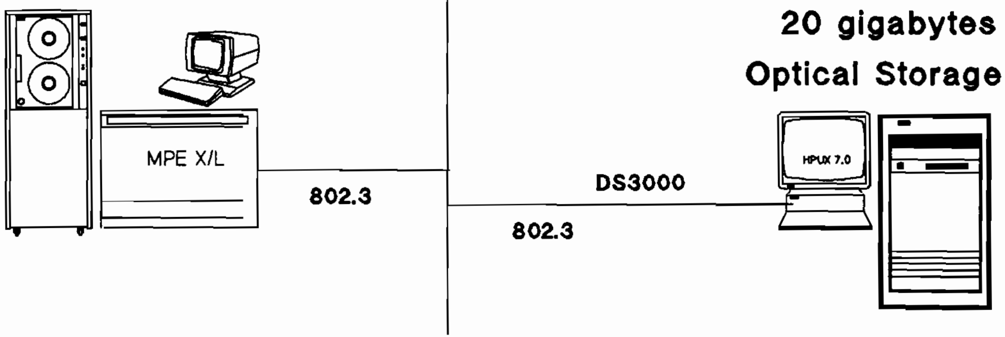
Transaction Logs

Audit Logs

Retains HP3000 File Attributes

20 gigabytes

Optical Storage



Client/Server Computing Environment

- Enabled by the growth of Networking individual processing units
- Must conform to standards of communications
- Used as a means to share large expensive computing resources
- Ideal Environment to distribute resources

  that are necessarily large

  that is expensive

  needs centralized management

Performance Management
in the
Distributed Computing Environment

Tony Engberg
Hewlett Packard

Definitions

Distributed computing environment: An environment which makes the needed resources and information contained within the network available to the appropriate users as though they were local to the users' workstations.

Distributed computing environment: A networked computing environment in which, literally, the network is the system.

"The network is the 'system'".

Introduction

Let's start with an essertion: DCE is an evolving reality of major importance in the computer industry. It is a focal point for research at universities and within the laboratories of the leading computer manufacturers. It has been, and is being, implemented in a variety of forms by the information systems departments of major corporations and government agencies. It is the subject of books and articles. Industry analysts, academicians, and corporate executives agree that DCE is here today, and that it will have a sharp impact on the way computers are acquired and utilized in the foreseeable future. It's interesting to note that these same individuals don't necessarily agree on the details of the definition of the distributed computing environment; they just agree that it's a reality that is here to stay.

The disagreement over details exists because of the fact that the concept is still evolving. A powerful vision of its ultimate realization exists: a world in which computer users choose their systems and connect to tha DCE with the same lack of concern we show today when we purchase an appliance and plug it into an electrical outlet. Consider it! A world in which you were able to select the best system for your particular job, connect it to a network without worrying about such things as compatability, and immediately derive all the benefits of the networks resources (hardware, software, and information) insofar as the security and access rules allow. Its a multivendor world, built upon standards and a flexible, extensible, architecture that allows the seamless, transparent integration of applications and hardware. It's a world that doesn't yet exist, but which is approximated in many university and corporate laboratories. More importantly, it can be found in nascent form in the network implementations of a number of major corporations.

It is not my purpose, here, to define and discuss the details of the DCE. I refer those desiring some grounding in the subject (or information on the debate it often engenders) to other sources [1]. I will further assume that the reader either believes my initial assertion, or finds it to be sufficiently plausible to be interested in examining one of the areas of development that will be critical to its success. This paper focuses upon the crucial subject of **performance management** within the DCE. It addresses several key questions. What is performance management? What issues exist, today, in terms of performance management in the area of distributed computing. Given the fact that the DCE is evolving, where is performance management headed? Please note at this point that I do not pretend to be a visionary; things get somewhat hazy for me the farther out I look. This paper will provide a clear depiction of the present, and a fuzzier image of the future.

Performance Management

A simple, straightforward, and very useful model of performance management exists today. The model divides its subject matter into four segments based upon the focus of each segment. Table 1 lists each segment along with a brief description of its focus.

	React.	Proact.	Focus
Diagnosis	x		Troubleshooting
System management	x	x	Managing existing resources to maintain system performance; Setting user expectations, and managing to these (e.g., via service level agreements)
Application optimization	x	x	Designing applications that are efficient and meet performance requirements; Tuning applications
Capacity planning		x	Planning for the resources required to meet performance goals in the face of business and environment changes

Table 1

Whether you are concerned with a single system or a network of systems, one operating system platform or five, there exists a sat of performance metrics associated with each segment. There are also sets of resources to be measured, and analyses to be performed. The model forms a good basis for distinguishing the different focal points of performance work, and for discussing the varying levels of complexity which different platforms and "systems" entail. In the context of the DCE, it also provides a vehicle for examining the role that each segment plays in achieving performance transparency, the primary concern of performance menagement in the DCE.

Performance Transparency

I know that I stated that it was not my intent to get into the details of DCE, but the concept of transparency is a critical one that cannot be ignored. Transparency exists when the "system" that is the DCE looks like a single system to the end user and application programmer. There are numerous types of transparency, such as location transparency (objects are accessible without knowledge of their location) and access transparency (identical access routines will get you to local and remote files). All are crucial to providing overall transparency.

Performance transparency has been defined in several ways. One popular definition is that it "...allows the system to be reconfigured to improve performance as loads vary." [1] I find this definition somewhat limited. It actually sounds more like a definition of scaling transparency or extensibility than of performance. A more apt description of performance transparency would characterize it as enabling the management of resources so as to maintain predictable, stable levels of performance. In other words, be it the location and removal of bottlenecks in a reactive fashion, or proactive tuning and capacity assessment, performance transparency results when your management of the "system" (automated or not) keeps the user populace, and application programmers, from knowing, or needing to know, what the "system" looks like and where the requisite resources exist. It is a crucial factor in the success of the DCE. There is nothing inherent in a "system" of systems and interconnections that makes unpredictable response and insufficient throughput any more palatable than they are in the single system environment. There are a lot of things, however, that make achieving performance transparency difficult. Every segment of the performance management model comes into play when attempting to achieve it.

Added Complexity

The step from a simple world of direct connect terminals and networks in which users (or, at least, application programmers) must know about the network map and the location of information to the world of the DCE is a large one involving extensive change. It should be glaringly evident to all that there are some fairly complex issues, not encountered in the optimization and management of single systems, that are inherent in the DCE. Let's consider just a few.

1. Levels of granularity

   What is the system you are attempting to manage? Is it a single system, a node, a cluster, or the entire DCE? It could be any of these, depending on what you're trying to do. You need to be able to determine what level you need to attack, and how to do it.

   There is another issue. If you are attempting to overlay the performance management model onto some model of distributed computing architecture (e.g., the NewWave Computing Architecture), what level or levels of the latter model do you need to focus on in order to solve your problem? Should you be examining one layer, all layers, or some in between set?

2. Component interactions

   When attempting to manage performance in the single system context the interaction of applications, operating system modules, subsystems, and the hardware, all must be considered. Measurement instrumentation must be in place to allow the identification of all of the components in order to permit effective analysis. Consider the problems that arise in a distributed environment when attempting to identify which components were interacting. First, components such as processes must be identifiable. This introduces the need for extensive instrumentation that is, currently, unavailable. Second, the ability to synchronize measurement information for analysis purposes must exist, without encouraging large amounts of overhead. Although the post processing of logged data might be considered a possible solution to this last problem, the stated need for "on-line, real time" diagnostic capabilities makes it a real challenge for at least one segment of the model..

3. Differentiators

   The DCE, as we've described it, is a multi-platform, multi-vendor world. This means that those charged with managing performance must either have a thorough understanding of the architectural and implementation differences (both glaring and subtle) that exist, or must have a set of measurement, analysis, and (potentially) optimization tools available that permit them to ignore these differences. Automated optimization in the DCE is, in my opinion, some way off; we haven't yet figured out how to optimize our single system solutions in a fully automated way. It is also, in my opinion, unreasonable to assume that our first pass at measurement and analysis tools will be able to fully relieve the user of the need to understand all of the platform and vendor distinctions.

It is, however, essential that these first pass tools go a long way towards meeting this need. Whether the knowledge resides in the head of the performance manager, or in the tool she or he uses, the complexity introduced into the act of analysis is high.

There are, obviously, many consequences of the increase in complexity. Perhaps the most important, on the surface, to the performance manager is the potential increase in the resources required. In automating the performance management process, for any particular segment, the amount of processor, memory, and disc consumed could be immense. The amount of knowledge and analytic talent required of the performance manager could also increase greatly. Approaches need to be developed which will minimize this effect of added complexity.

Performance Management in the DCE Today

Diagnosis

Diagnosis can be generally defined as the investigation into, or analysis of, a condition or problem, or a conclusion drawn from such an analysis. Performance diagnosis is a reactive activity, occasioned by the perception of a problem. It is often, although not always, a real-time pursuit. It leaves the realm of real-time fairly frequently. How often have you triad to respond to a problem call only to discover that, by the time you were in a position to start examining the problem, the source of the trouble had disappeared? In the case of transient, recurrent problems, some form of historical analysis is required.

Today, the ability of the performance manager to diagnose problems within the DCE is sharply limited. The restrictions include:

1. Current measurement and analysis tools provide excellent single system views, but no DCE level insight. Thus, the best that a performance manager can do is to collect information from each system, and traffic data from the network, and then try to put all of this together in order to find the source of the problem.

2. Diagnosis today is extremely lebor-intensive, due to the fact that it entails the use of multiple tools, hand synchronization of data, and a very deep understanding of the specific "system" with which the performance manager is dealing.

3. In situations in which measurement and analysis tools support multiple operating system platforms (still within a single system context) and allow comparison of those platforms from within a single display mechanism (e.g., LaserRX) there is a single vendor limitation. True multivendor troubleshooting takes us back to point 2.

4. Diagnosis requires a great deal of resource, primarily I/O,

CPU, and disc space, when a problem is sufficiently knotty to require lengthy traces and a great deal of data correlation. In the DCE, "lengthy traces" may result from only a few minutes of in-depth monitoring, depending on the measurement tools in use. The processor power required to correlate and analyze this data can also be large.

5. The lack of a standard measurement interface across platforms and vendors, including any type of synchronization mechanism, makes the data that can be obtained and correlated of limited value at best, and of no value at worst. The performance manager is put in the unenviable position, often times, of having to look for the needle of unknown correlations that will point to a problem in a haystack of raw, unsynchronized, data.

In short, diagnosis within the DCE today is lagging the needs of those charged with managing performance. What must be done to remedy this problem will be discussed below.

System Management

The system management segment of performance management involves the management and optimization of the performance of the "system" given a set of existing resources that define that "system". It includes setting user expectations via devices such as service level agreements, monitoring performance and verifying that those expectations have been met, proactive problem solving, and various types of management reporting. It suffers from many of the shortcomings of measurement and analysis that plague the diagnostic world, and then some.

1. The "system" manager is limited to being a system manager; the single system level of scrutiny is all that is really supported at present. Network information is available, but not in a form that can be readily correlated with overall performance activity.

2. The measurment and analysis solutions that are available that are single system/multi-platform with the ability to display numerous system concurrently are single vendor solutions.

3. Measurement and analysis in this environment is extremely resource intensive (both human and machine).

4. While expectations can be set through such time-honored methods as experiment, intuition, or darts, and the data can be gathered to verify what response time and throughput was actually seen by the users of a specific system, the actual performance is not controllable in any real sense. Problems cannot be anticipated easily (or at all in most cases).

5. Bottleneck analysis is extremely limited, although the

collection of data on single systems with a tool such as LaserRX, and the export and correlation of that information by a user-created program could be used to locate some impending bottlenecks and their causes. Again, the resource investment is high.

On the plus side, load balancing and job scheduling is available for use in many distributed environments. HP's Team Computing provides access to both of these optimization aids for performance managers.

Application Optimization

Software Performance Engineering (SPE), aimed at "designing performance in" to software, hardly exists today in most development shops. In the area of DCE, it is essentially nonexistent. This is not a slur on the good name of application developers; rather, it is a statement about the huge number of unknowns in the evolution of the DCE, about the dearth of tools, rules of thumb, or standards available, and about the difficulty of building true multiplatform/multivendor applications today.

What about tuning tools, wholly reactive in nature, for fixing applications once they are in service. Again, the list of such tools is, to the best of my knowledge, empty, unless one includes single system tuning devices (e.g., GPROF, SPT/XL). These do not, however, truly qualify as DCE application optmization tools.

Capacity Planning

Capacity planning is an exercisa in business forecasting. Its objective is to ensure that, as business needs and demands change over time, as system workloads grow or contract, an acceptable, cost-effective level of system service is provided, **and can be planned for in advance of its requirement**. This last clause is an important one. Capacity planning is a proactive activity; if the diagnostician is delivering the message that the system is "out of gas", then the capacity planner has failed.

What is required for the capacity planner to succeed? "Understanding the relationships between business requirements, computer workload, computer capacity, and the level of service or responsiveness required by users is the key to effective capacity planning." [2] Capacity planning is business focused, and directly concerned with the effects of impending workload changes on critical performance metrics.

Capacity planning, today, is normally carried out in one of two ways. Many sites utilize statistical forecasting methods to extrapolate future reqirements from past experience. This method works quite well provided one makes some basic, obvious assumptions about the manner in which one's workload is changing and about the bottlenecks that will be encountered as a result of this change. Statistical forecasting also permits the mapping of user-specific business units to system metrics, a handy device if carefully applied.

The second method used for capacity planning today involves the representation of the system and its workloads via analytic models which can be used to answer "what if" questions concerning workload and system configuration changes. Although significantly more complex to utilize and to interpret than statistical forecasting techniques, analytic modeling provides a degree of flexibility and a depth of analysis not found in the former approach.

Given the above, where do we stand in terms of capacity planning for the DCE? In a very uncomfortable position, unfortunately. The complexity of the "system" goes far beyond that of the systems to which we apply the techniques mentioned above. The basic assumptions of statistical forecasting make its applicability extremely limited (almost to the point of irrelevance) when considering the impacts of change in the "system's" workload (although this is still a powerful tool for examining each component system). Further, the complexity of the DCE is such that it does not yield to our present analytic modeling approaches. To put this another way: the ability to determine the effects of another disc, or a faster processor, on a single system's throughput or response time does not give us the ability to determine the impacts of adding a file server or mail server to a set of workstations in a DCE.

Work is, of course, being done in this area, and interim solutions are in use. HP works very hard to characterize the performance of specific configurations of clusters (what one might call "DCE packages") of its systems, and to provide users with guidelines in order to set expectations and permit some estimation of impacts. Other vendors also attempt to provide some characterization of their packages. The limitations of this approach in a competitive world is obvious, however. The impossibility of a user attempting to do her or his capacity planning via any type of benchmarking within a DCE should also be evident.

Another approach is simple "back-of-the-envelope" modeling. This is an extremely labor intensive exercise in a DCE of any size, for the measurement and analysis readings discussed in early sections. The number of variables to be controlled can be unwieldy, if the analyst can even determine which variables are critical to the prediction of performance. Note that the variables that may be crucial to one decision may be markedly different from those which are crucial to a different decision. The difference may be as simple as a change in configuration or in the workload on a specific server.

Where Do We Stand?

The above description of the current state of affairs, sketchy as it is, should make several points clear. First, the newness of the DCE concept, combined with rapidly advancing technology, has produced a degree of uncertainty in several areas, of which performance management is one. This uncertainty is having a marked effect on the speed with which R&D efforts can proceed.

Second, in those places where a sustained investment is being made to produce the technology required by performance managers the added complexity introduced by the DCE is requiring a significantly larger investment of resources than other recent advances, such as RISC, multiprocessing, etc.

Third, the primary means of managing performance in the DCE today are the same as those available in the single system environment, and this leaves the performance manager far short of her or his target. The current technology and methodology can only be applied to a limited extent, and such application is extremely labor and machine resource intensive.

Performance Management in the DCE

If the DCE is to succeed, and there is cogent evidence that the corporate world believes in it and will not allow it to fail, there must be tools, techniques, and analysis methods made available that will support those choosing to implement "systems". HP's Performance Technology Center is focusing on providing these necessities. Although there is not room here to go into all of the work that has been, and is being, done, several observations about the future of each segment of the performance management model can be made.

Diagnosis

In a situation in which the needs of every segment are critical to the ability of the performance manager to do his or her job, this segment is the most critical. With any new operating system, architectural advance, or major application, diagnosis is the first requirement. The history of diagnosis is rife with hack tools and labor intensive analysis solutions in which performance managers are presented with reams of data, often unvalidated or of no interest to those outside the vendor's OS and hardware labs, and are expected to shoehorn these tools into solutions for their very specific diagnostic problems.

Diagnosis in the DCE will require the ability to quickly isolate bottlenecks and "performance faults" automatically (via such things as alarm mechanisms). Pinpointing problems to specific "components" of the "system", be they hardware or software, at high or low levels of granularity, will be essential. The ability to do this in a manner that does not consume excessive resource, that permits management of all or parts of the total DCE from a single point, and that allows multi-platform (indeed, true multi-vendor) analysis will be critical.

System Management

Resource utilizatio will have to be tracked across the DCE, as well as across specific subsets if the overall performance management task is distributed. This means, again, that resource constraints (at the global, application, process, and "object" levels) will have to be dealt with. The formation, reporting, and enforcement of service level agreements in the DCE will require automated control and tracking mechanisms. Management and reporting of data will have to be, as indicated, scalable. Automated balancing and routing mechanisms will have to be put in place. True multivendor support will have to be implemented.

Application Optimization

Application activity, at many levels, will have to be tracked across the DCE. This will mean a level of identification not available in even the simple distributed systems' applications currently in use. Reactive tuning work will demand that this ability be automated, that it not require excessive resource, and that offending applications be quickly "tunable". The tie to the previous two segments should be evident. Diagnosis may well point out an application performance problem; system management will demand that mechanisms be available to quickly adjust the DCE to control the offender, and application optimization will require that the problem be quickly repaired.

On the proactive side, the need will exist to characterize the application's interaction with the DCE at some level (granularity enters the picture here), and to feed information regarding the interaction into the capacity planning tools. Other needs will exist which can be classified as SPE/CASE related.

Capacity Planning

This may well be the toughest problem of all for the performance manager. Capacity planning will have to be delivered at many levels, and it will all have to fit together. Individual systems will require evaluation, as will nodes, clusters, etc., up to the total DCE. The complexity will be high, characterization of workloads at the upper levels will be extremely difficult. This, truly, will be the challenge of the early 90's for those called upon to deliver performance management technology.

Looking Ahead: Final Issues

Meeting the performance management needs of those who are building DCE's today and those who will be implementing them tomorrow is a monumental task. It can be done, if certain critical pieces of the solution are provided by the vendors who are producing the components that will make up the DCE's of the future. These requisite items can be stated quite simply.

First, standards in general, and standards for measurement in particular, will have to become a reality. The battle over standards will have to end; some fairly significant advances will have to be made rapidly. Many might argue that this is asking for the impossible; standards and rapidity don't mix. However, a great deal of work has been done in this area already, via a working group made up of members of the Open Software Foundation (OSF), Unix International, and several "impartial outsiders". This working group, of which HP is an active member, is preparing a submission for performance management measurement standards to POSIX. This is a very good sign; it indicates that there is an excellent chance that we will have the basis for a real multivendor solution.

No standard is of any use if it is not implemented, and in the case of a measurement interface "implemented" must be understood to include validation. This is the second requirement if we are to succeed. Vendors must be sincere about providing standardized measurement interfaces that have been validated, and this means making a significant investment beyond that being made today. Consider the size of the task. Today we have, at best, instrumentation within part or all of various operating systems modules. Tomorrow we will need instrumentation within all system network, and object management services. It's a large task.

A third requirement will be the provision of the necessary measurement facilities in conjunction with the release of the services that they are intended to monitor. If this is not done then those attampting to provide the technology required for performance management in the heterogeneous distributed resource environment will be unable to provide complete solutions to those charged with the task of diagnosing performance problems, monitoring and managing "system" performance, optimizing applications, and planning for growth.

Those who are designated as the performance managers for DCE's over the next few years will live in interesting times. Contrary to the old curse, this does not have to mean suffering. It will require that individuals taking on this task be comfortable with the basic concepts of performance, be attuned to the needs and realities of their businesses, have an analytic bent, and be willing to work with vendors to ensure that the appropriate technology is made available in ordar to implement all segments of the performance management model. It will take individuals who are not afraid of change, who enjoy experimentation, and who understand the need to establish and refine processes over time. For those who fit this description, the 1990's should be a source of professional growth and, most importantly, should be enjoyable.

References

[1] Distributed Systems Concepts and Design, Dollimore, et. al., 1988 Addison-Wesley, Wokingham

[2] Capacity Planning: A Primer, alpha test preliminary version, 1990, Hewlett Packard Co.

Building a Network Management Solution

Steve Richard
Hewlett-Packard
19420 Homestead Road
Cupertino, CA 95014

In the 90's networks will grow even more complex and geographically dispersed. As a result, organizations will face a rapidly increasing need for effective network management. Companies can attack this problem in the most cost-effective way by beginning to build an integrated network management solution. Hewlett-Packard provides HP OpenView, a family of network management products which serve as a solid, open foundation on which to start building.

This paper will take a technical look at how the new UNIX-based HP OpenView Network Management Server product can be used to build a network management solution. A hypothetical solution will be developed using a real network scenario.

First, the six basic steps for developing a solution are covered: evaluate the environment to manage, define objects, define system structure, define object managers, and define applications. Going through these steps will generate a list of specific applications and object managers that need to be implemented.

The second part of this paper will cover some of the details of programming one of the applications needed for the solution. We will look at the application's usage of the programmatic interfaces, its basic structure, and some of its actual code.

Designing a network management solution is like building a house. You need a good foundation, especially for a large house, and you need many pre-made materials like doors, windows, walls, etc. Each family has unique needs and thus the overall design of the house needs to be custom. Network management is the same way. Using the Network Management (NM) Server as a foundation, a unique solution can be built that meets the needs of a specific network environment. There will be functionality available to manage specific objects that are like the parts of a house. They need to be integrated into an overall design or solution.

The process of building a network management solution will be illustrated in this paper by addressing the needs of an extended LAN environment at a HP Division. In other words, how can the HP OpenView products be used, especially the HP OpenView NM Server, to build a network management solution for a HP Division LAN? Much of the material for this paper comes from interviews with the administrators of one of the HP Division LANs. In effect, I acted as though I had been hired to design a custom network management solution for them. The resulting solution is intended to be realistic and cost-effective. Through the remainder of this paper the LAN used as an example is referred to as the HP Division LAN.

Description of Network

The Network shown in Figure 1 is made up of four subnetworks. The subnetworks are called the Hands-on Network, the Office Network, the Timeshare Network, and the IBM UNIX Network. Each of these four subnetworks is connected to each other and to the rest of the Internet via a Cisco Router. The Office Network is broken into three segments with two LAN bridges.

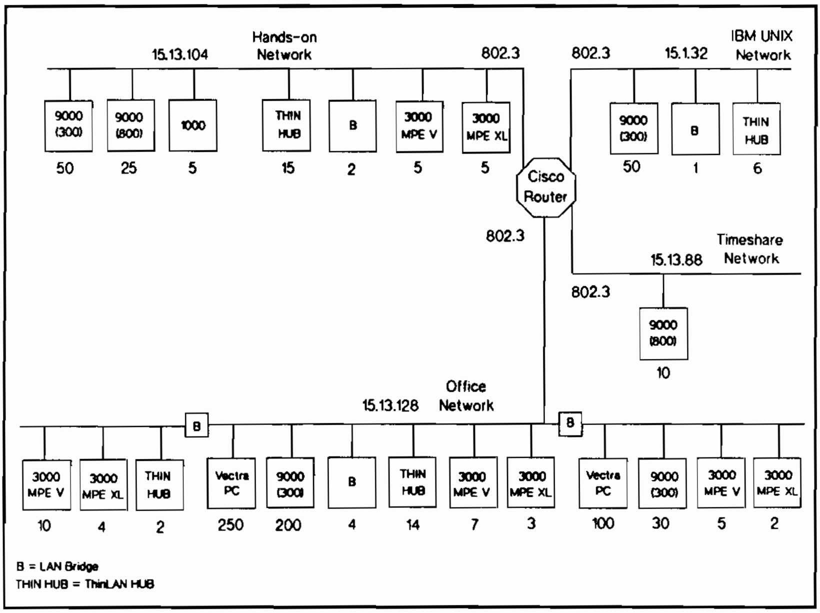


Figure 1. IND LAN

The network connects HP 9000 workstations (Series 300s) and multi-user systems (Series 800s), PCs, HP 3000s (MPE V & XL) and HP 1000s. To accommodate the communications there are LAN bridges, 1 gateway, repeaters, Thin LAN hubs, and LAN segments.

Before we look at how to solve the management problems presented by this network, let's meet the second component of the puzzle—The HP OpenView NM Server.

How to Use the NM Server

The NM Server is our greatest ally for resolving the management problems of today's complex networks. To build a solution using the NM Server we need to understand what it is and how it can be used. The NM Server is an open standards-based environment for integrating distributed network management functionality. You can think of the NM Server as an operating system for network management. There is a Developer's Kit and a Run-time product. The Developer's Kit is a fully documented set of application programmatic interfaces (APIs) and tools for developing network management applications. Developers can leverage from existing services provided by the NM Server and from the services provided by other developers.

The HP NM Server is a realization of HP's OpenView Network Management Architecture that is based on OSI standards. The OSI standards include an architecture model, a set of services built on a standard protocol, and a way of defining information about *managed objects*. The model defines application processes called managers on management nodes and application processes called agents on managed nodes. Managers direct agents to perform operations on managed objects for which they are responsible. The standard services and protocol are CMIS (Common Management Information Services) and CMIP (Common Management Information Protocol). CMIS are a set of services the manager or agent process can use to exchange management information between them. CMIP is the protocol exchange between the manager and agent process when transferring management information. Information about *managed objects* is specified through the SMI (Structure of Management Information) and the MIB (Management Information Base). The SMI is a set of rules on how objects should be defined and the MIB is a virtual collection of all defined objects.

The NM Server has four main components:

- The Communications Infrastructure
- HP OpenView Windows (integrating user interface)
- Data Management Services
- Event Management Services

The features of each of these components are shown in Figure 2.

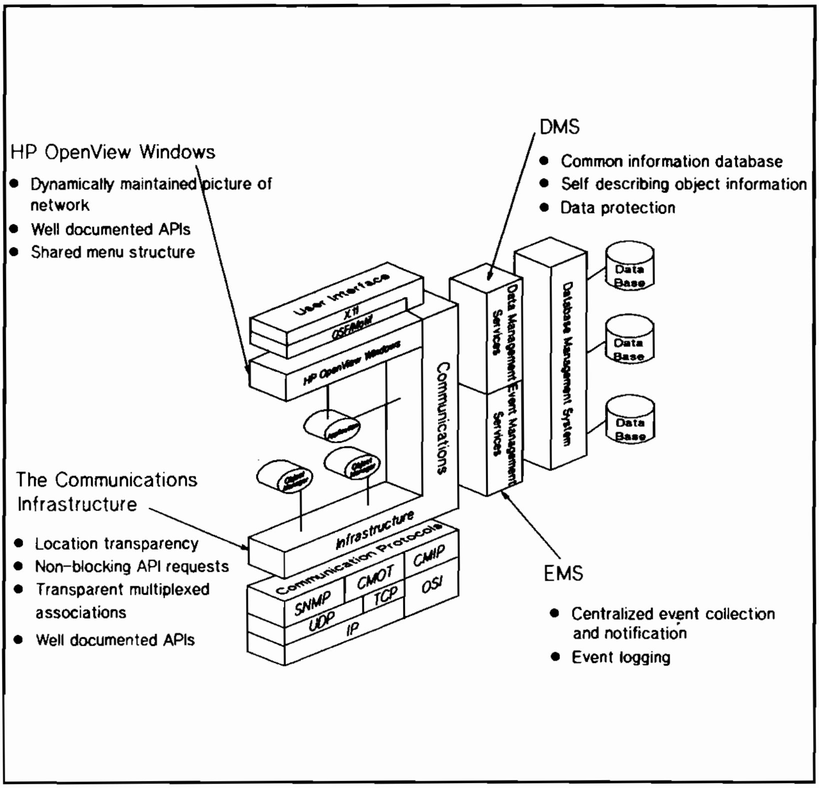


Figure 2. NM Server Features

The Communications Infrastructure provides transparent access to distributed network management services. The Communications Infrastructure includes industry leading implementations of the Simple Network Management Protocol (SNMP) and the Common Management Information Protocol Over TCP/IP (CMOT). These are standard protocols in the Internet Community for transferring network management information. HP's SNMP is fully interoperable with existing implementations and includes features above and beyond the industry standard. All network management services are accessed through the Communications Infrastructure API (OVc API). This means that SNMP, CMOT and later the Common Management Information Protocol over OSI (CMIP) are accessible through one, common interface based on CMIS.

Data Management Services provides transparent access to a single logical repository for storing data. This eliminates the possibility for inconsistent data in the network because of differing values in multiple databases.

Data Management Services can actually be used for three important functions. The first and most common use of DMS is to store versions, or snap shots, of an object's data. DMS also can be used when doing testing or prototyping of applications by acting as the manager of the data your application needs to access. A third use of DMS is to develop the ability to handle multiple types of network objects with the same code. You can develop this type of functionality by accessing information about network objects from DMS (MIB information). This information fully describes the characteristics of each manageable object in the network. DMS uses this feature itself to store versions of any object type with the same code.

Event Management Services provides a centralized way to generate, receive, route, and log network events. EMS is like a post office for network events, but the letters sent to it are not addressed. Instead, applications tell EMS which types of letters they would like to receive.

HP OpenView Windows is used to integrate the user interface for all network management functionality. HP OpenView Windows manages the set of menus and graphical map of the network which all applications share. HP OpenView Windows makes it seem to the user like there is one large integrated application for managing the network. HP OpenView Windows is built on X-Windows and OSF/Motif.

Designing a Solution Using the NM Server

Network management solutions are developed in six steps:

1. **Evaluate the environment to manage.** Identify and characterize the important hardware and software in the network and what the network management needs are for the network.

2. **Define Objects.** Identify the objects that need to be managed (based on the needs) and what their properties are.

3. **Define System Structure.** Define all the components of the network management solution and their relationships.

4. **Model Objects.** Formally model the exact attributes and properties of each object.

5. **Develop Object Managers (Agents).** Formally design the object managers that need to be developed. An object manager is equivalent to an agent in the OSI model. An object manager is a piece of software that represents an object that you want to manage. This step includes deciding on the exact functionality and structure of each object manager.

6. **Develop Management Applications.** Formally design the applications that need to be developed. Applications use the services provided by object managers.

Now, using the HP Division LAN as an example, we will go through each of these six steps in detail.

Evaluate the Environment

The first step in designing a network management solution is to evaluate the environment. This step involves asking the following questions:

1. What hardware and important software is on my network?

2. What are the most common or time consuming problems related to managing my network?

3. What is currently done and what could be done to deal with these problems?

The result of this step is a clear and common understanding of users' needs for network management functionality.

In our example, The HP Division LAN contains the following hardware and software:

- 350 HP Vectra PCs
- 330 HP 9000 300 Series (workstations)
- 35 HP 9000 800 Series (multi-user systems)
- 27 HP 3000 (MPE V)
- 14 HP 3000 (MPE XL)
- 12 LAN Bridges
- 1 Gateway
- Repeaters
- Thin LAN Hubs
- LAN Segments
- NS Software
- X.25 Link Software (12 systems)
- LAN Link Software

This information can be used to help prioritize the needs related to each object and give a big picture of the network.

Identification of Problems

The next part of evaluating the environment is to identify the most common or time consuming problems. Interviews with the administrators of the HP Division LAN revealed the following common symptoms and problems.

Symptoms:

- Unable to communicate between two systems
- Poor performance

Problems:

- Don't know what's out there
- Duplicate IP addresses
- Run out of bandwidth
- Configuration problems on individual systems

The next part of the evaluation is to find out what is or can be done to prevent, detect, or isolate the problems.

Unable to communicate between two systems can be:

- Prevented by eliminating duplicate IP addresses.
- Prevented by maintaining correct address information. Need to be notified when there is an address change or addition
- Detected by doing "Pings" on critical systems automatically. A "Ping" is a link-level echo test between two systems that is available on HP 9000s.
- Detected by monitoring the error logs on critical systems
- Detected by automatically checking status of critical systems
- Isolated by viewing summary status of systems
- Isolated by doing a series of testing that answers the following questions: Is the network down? Is the system down? Is it a resource problem? Is a segment or component down? What layer?
- Detected by checking status of ThinLAN Hubs

Running out of bandwidth can be

- Prevented by monitoring and trending the bandwidth utilization of each LAN segment, the router link, the X.25 link, the X.25 gateway, and the bridges

Poor performance can be

- Prevented by monitoring and trending important utilization of each LAN segment, the router link, the X.25 link, the X.25 gateway, and the bridges
- Detected by detecting broadcast storms
- Isolated by knowing the distribution of network use
- Detected by knowing response times

Not knowing what is out there can be

- Prevented by having a graphical representation of the network that includes all devices

Define Objects

The second step in designing a network management solution is to define objects. This step involves asking the following questions:

1. What physical, logical, and functional objects are involved in performing the preventing, detecting, and isolating identified in the previous step?
2. What, roughly, are the data, functions, and events that need to be managed by an object manager for each object?

The result of this step is a clear understanding of what network objects need to be managed and how.

By going through the list of ways to prevent, detect, and isolate the common problems, the objects listed below are identified. The functions, data, and events associated with each object are also shown. An example of a piece of data is an error counter. An example of a function is to reset an object. An example of an event is a notification that an object is shutting down.

HP 9000: (critical systems)	function: "ping" data: status data events: from error log data: resource data
HP 3000: (critical systems)	events: from error log data: status data data: resource data
Bridges:	data: to calculate utilizations data: status data data: resource data events: change in status

Hub:	data: status data
Router:	data: to calculate utilizations of itself and line data: status data events: change in status
LAN Segments:	data: to calculate utilization and distribution events: detecting storm
X.25 Software:	data: to calc utilization
Functional Objects:	function: Automatic Pings function: Monitor IP Address Information function: Graphic Representation of Network data: utilization data (with trending)

Define System Structure

In order to design an effective system structure, you need to better understand the object oriented nature of the NM Server environment. When using the NM Server, all network management software needs to be either an application or an object manager. An object manager is a piece of software that represents a network object such as a system, modem, or stack layer. Using an object manager is somewhat like linking in a library to an application. Any application can use any object manager's services. An object manager hides the complexity of dealing with the object that it represents. This division of software promotes reusability of object related software and allows applications to be developed more quickly and with more power. Figure 3 shows the differences between applications and object managers in the NM Server environment.

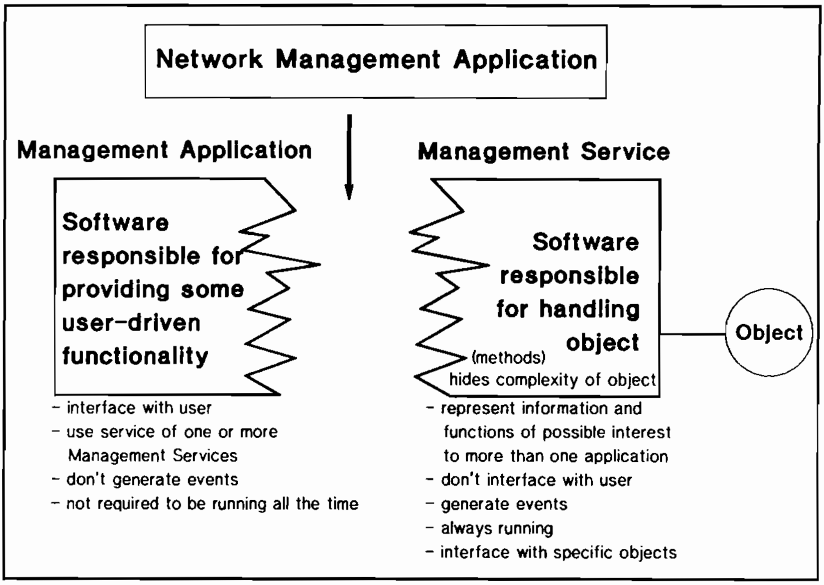


Figure 3. Applications vs. Object Managers

Defining the system structure is the third step in designing a network management solution. This step involves asking the following questions:

1. What existing object managers and/or applications can be used?

2. What object managers and applications need to be implemented?

3. What are the relationships between the object managers and applications and with the NM Server?

The result of this step is a clear understanding of what the components of the solution will be and how they will interrelate.

The following is existing network management functionality that relates to the objects that we want to manage in our example. All pertinent functionality is listed independent of whether it integrates into the NM Server environment. This is done because depending on the requirements it might make sense to use a non compatible product at least for the short term.

The HP OpenView Node Manager (will run on the NM Server) - provides fault, configuration and performance management of multi-vendor TCP/IP networks. The Node Manager runs on HP 9000s and manages network devices that are IP addressable and/or run the SNMP protocol. The Node Manager functionality includes the ability to show real-time network connectivity, display statistics, check connectivity between two nodes, and display resource information.

The HP OpenView Bridge Manager (runs under HP OpenView Windows on a PC) - provides the ability to centrally monitor and control HP LAN and StarLAN Bridges in an extended IEEE 802.3 or Ethernet LAN environment. It provides facilities to monitor network traffic conditions, help diagnose network problems, enhance network security, and increase network reliability.

The HP OpenView Hub Manager (runs under HP OpenView Windows on a PC) - provides the ability to centrally monitor and control HP ThinLAN Hubs in an extended IEEE 802.3 or Ethernet LAN environment.

Access to the Router via the SNMP Protocol - The router in the network has SNMP implemented on it which will allow it to act as an object manager for use by OpenView management applications.

Access to the Bridges via the SNMP Protocol (planned) - The LAN bridges can be upgraded so they have SNMP on them. This would allow them to act as object managers for use by OpenView management applications.

Access to the Hubs via the SNMP Protocol (planned) - The ThinLAN hubs can be upgraded so they have SNMP on them. This would allow them to act as object managers for use by OpenView management applications.

The HP LanProbe (runs on Microsoft Windows on a PC) - LanProbe is a distributed network analysis and monitoring system that is protocol and vendor independent. It monitors the LAN continually and analyzes everything it sees. It displays on one screen the real time segment traffic in terms of number of packets, LAN utilization, bytes, broadcasts, errors, and collisions.

Our network management solution can be built primarily from the functionality listed above. The Node Manager is well suited to the type of network we want to manage and will be the cornerstone of our solution. The Node Manager will allow us to access all the information we are interested in on the HP 9000's and the Router (which has SNMP on it). It will also automatically check connectivity, monitor for duplicate IP addresses, and do some problem isolation between nodes.

To get the status and utilization data from the Bridges and Hubs we need to have either a second PC display for their management software, wait for this software to be available on the NM Server, or upgrade them to the latest models which support SNMP. For the HP Division LAN, I chose to upgrade the devices since integration is very important and the cost of upgrading will pay off over the long term. With SNMP, the data on these objects can be displayed by the Node Manager.

To monitor the utilization of the LAN segments we need a product like the HP LanProbe. Unfortunately, there is no way to integrate this functionality with that on the NM Server so we will need a separate PC display for this.

A few applications and object managers need to be implemented to complete the solution. We need to implement object managers for the following:

- accessing data on the HP 3000
- for managing network utilization data
- for managing the X.25 links

We need to implement applications for the following:

- displaying object utilization in a standard way.

The system structure for the solution is shown in a hierarchical form in Figure 4. The figure shows the applications that would sit under HP OpenView Windows and also the object managers that each of the applications makes requests on. The applications are shown as circles and the object managers are shown as rectangles.

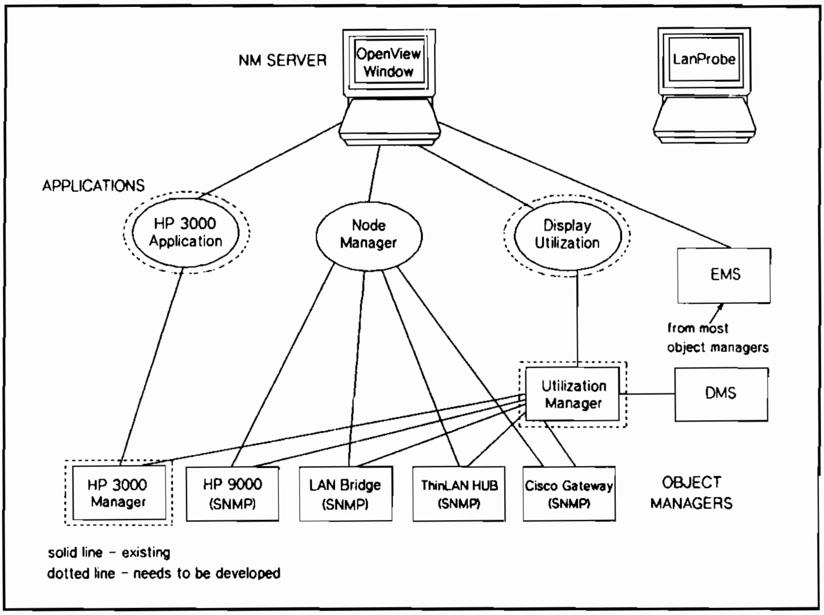


Figure 4. The System Structure

Model Object

The fourth, fifth, and sixth steps of designing a network management solution are more detailed steps related to implementing components of the system structure that have been defined. For these steps we will narrow our focus to implementing a specific object model, object manager, and application. We will look at the fourth and fifth steps using the HP 3000 object manager and application as an example. Modeling an object involves asking the following questions:

1. What will the exact attributes (data to make available) of the object be (type, etc.) and what are their types?

2. What operations can be performed on the object (GET object's attributes, SET object's attributes, CREATE instance of object, DELETE instance of object, perform ACTION on object (like resetting it))? What information needs to be specified for each action?

3. What events will the object generate? What event information will each event pass?

Modeling an object involves a little more detail than this but these questions will generate tbe primary information needed.

For the HP 3000, the HP system administrators indicated that they would like to be able to access the following types of attributes:

- NS Services status ("nscontrol status=services")
- TCP/IP statistics
- internal tables (address cache)
- network interface status ("netcontrol status;ni=lan")
- link statistics

For simplicity let's just say we want to be able to access the lan interface data available from the :NETCONTROL STATUS:NI=LAN command. The attributes and their types are:

```
NETWORK INTERFACE STARTED : string
FLAGS                     : integer
NI PROTOCOLS              : string
MAPPING TABLE SIZE        : integer
MAPPING TABLE ID          : string
OUTBOUND BUFFERS          : integer
INBOUND BUFFERS           : integer
NETWORK INTERFACE TYPE    : string
NAME                      : string
TRACE MASK                : integer
IDLE DEVICE TIMEOUT       : integer
```

The 3000 object manager will support only get and set requests since a need for create, delete, or actions was not identified.

The LAN administrators indicated that they would like to be notified of the exception conditions listed below. These conditions show up as subsystem error messages that are printed to an HP 3000 console.

TRANSPORT NOT ACTIVE
REMOTE ABORT
LOCAL NS GOING DOWN, CONNECTION FAILED

Develop Object Managers

Before we look at how to develop the 3000 object manager and application, we need to understand the basic protocol that is used between application and object managers and the APIs that are used. Applications send requests to object managers which receive them as "indicates." If the application requested a response, then the object manager will send a response back to the application which receives it as a "confirm." This basic protocol is shown in Figure 5.

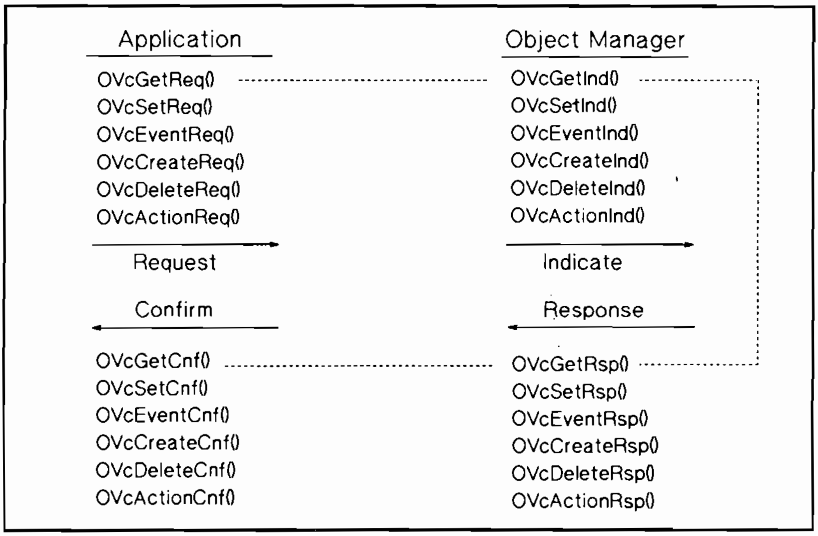


Figure 5. The Communications Infrastructure Intrinsics

Figure 5 also shows the intrinsics that applications and object managers actually use. There are intrinsics that correspond to each type of request that an application can make: get, set, create, delete, action, event.

The fourth step in designing a network management solution is to develop object managers. Since the functionality of the object manager has already been defined when the object was modeled, this step is primarily concerned with <u>how</u> to support the object definition. This step involves asking the following questions:

1. How will my object manager be structured exactly?
2. How will I get data for the object?
3. How will I generate events?

The result of this step is a fully designed or working object manager.

The basic structure of the 3000 object manager is similar to the basic structure for any object manager and is shown in Figure 6. The functionality of an object manager can be broken into two main pieces. On one side, the object manager needs to participate in the CMIS environment by servicing requests and generating events. This side needs to be developed very carefully to ensure interoperability. An object manager needs to act in accordance with CMIS specifications for the way requests are handled and the way errors are returned. The other side of the object manager needs to provide access to the object. This side of the object manager ideally should have an API that completely shields the complexity of dealing with the object from the CMIS half of the object manager.

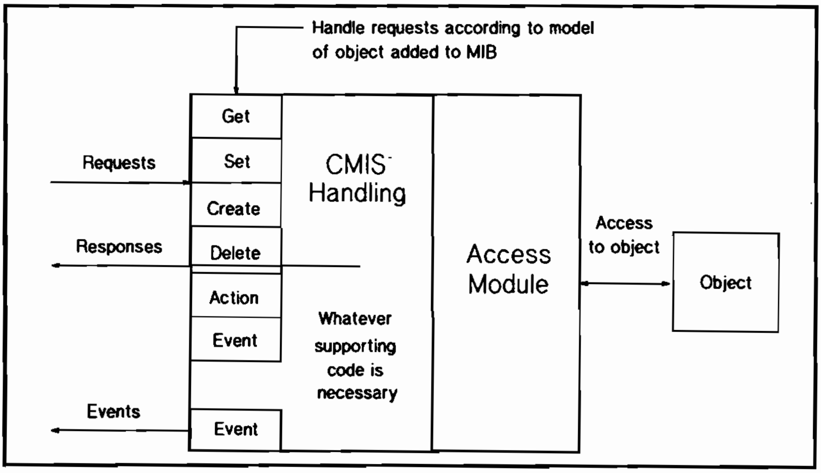


Figure 6. Basic Structure of an Object Manager

The detailed structure of an object manager is beyond the scope of this paper. The detailed structure of the CMIS half of the object manager is basically the same as for any object manager. There is a main loop that waits for messages of all types. When it receives a message from the Communications Infrastructure it calls a routine that will read the message in, etc. There is a routine to handle each type of request that will eventually be called. These routines use the access module to service the request.

Access Modules have no predefined structure. The access module for the 3000 object manager has a routine called "Get3000Data();".

Some of the main code for the 3000 object manager is shown below. The actual code is included only to give you a flavor of the programming. If you are not interested in the details, then reading the bold comments in the code is adequate.

```
/* wait for request */
Wait(readmask, &incoming_handle);

/* receive request */
OVcReceive(incoming_handle, OVc_READ, &cntl, sizeof(msg), msg);

/* make sure message is a request */
if (cntl.u.msg.primitive != OVc_PRIMITIVE_INDICATE)
    return;

/* save the request id */
req_id = cntl.u.msg.req_id;

/* decode the request */
opt_parms[0].keyword = OVc_OP_SCOPE;
opt_parms[1].keyword = OVc_OP_SYNC;
opt_parms[2].keyword = OVc_OP_FILTER;
opt_parms[3].keyword = OVc_OP_ATTRID_LIST;
opt_parms[4].keyword = NULL;
OVcGetInd(msg, &cntl, decode_buff, sizeof(decode_buff), &object, opt_parms);

/* make sure object exists */

/* get data using access module which uses MS */
Get3000Data()

/* build a CMIS attribute list to send back */
attrList = ...

/* send response */
opt_parms[0].keyword = OVc_OP_ATTR_LIST;
opt_parms[0].variable = attrList;
```

```
opt_parms[1].keyword = OVc_OP_CURR_TIME;
opt_parms[1].variable = currTime();
opt_parms[2].keyword = OVc_OP_MANAGED_OBJECT;
opt_parms[2].variable = object;
OVcGetRsp(bid, req_id, opt_parms);
```

Develop Applications

The fifth step in designing a network management solution is to develop applications. Developing applications involves using the Communications Infrastructure as well as OSF/Motif and HP OpenView Windows to convey information to the user and receive commands from the user. This step involves asking the following questions:

1. What functionality will the application provide?
2. What will the application's user interface look like (menu items and dialog boxes)?
3. Which object managers will the application use (DMS, EMS, 3000 Object Manager, others...)?
4. How will the application be structured exactly?

The result of this step is a fully designed or running application.

For the 3000 application we simply want to display the data on the 3000. The application will have one menu item called "Read Parameters" and will display a dialog box that shows the raw data on the 3000 (no graphs or anything). In order to service a user's request to display the 3000 data we need to use the services of the 3000 object manager. We do not need to use the services of EMS or DMS.

The basic structure of the application will be similar to that for any application which is shown in Figure 7. The application has modules for each menu item that it supports. The code to support each menu item can be broken down into two main sections—one to issue the appropriate requests and the other to receive the responses and display the data.

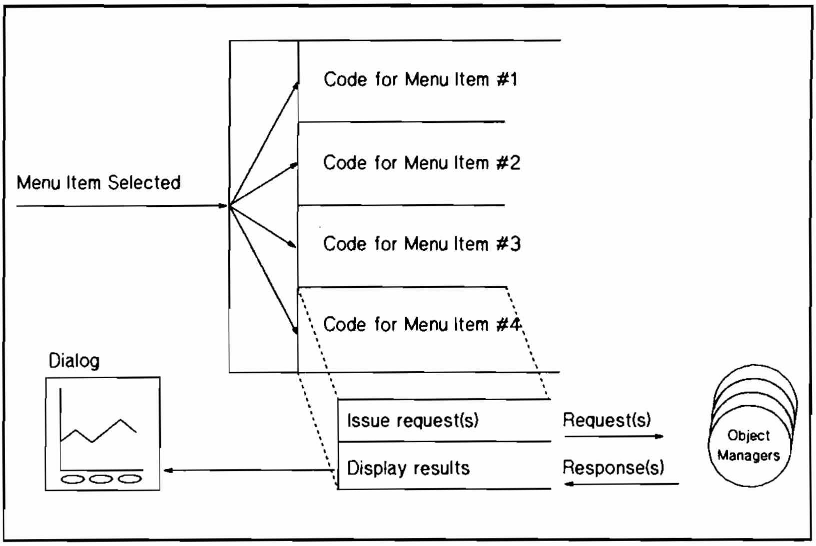


Figure 7. The Basic Structure of an Application

The following code would be executed after a user clicks on the "Read Parameters" menu item:

```
/* retrieve the last selected object on the graphical map */
/* from OpenView Windows                                   */
OVwSymGetNameInfo(selList[i].szName, szLabel, &objType,
szSubType);

/* retrieve data from dialog entry fields */
use OSF/Motif function calls...

/* request information from 3000 object manager */
opt_parms[0].keyword = NULL;  /* no optional parameters */
req_id = OVc etReq(bind_id, &object, opt_parms, &handle,
&status);

/* wait for response */
Wait(handle, &incoming_handle);

/* receive response */
OVcReceive(incoming_handle, OVc_READ, &cntl, sizeof(msg),          msg);
```

```
/* indicate which optional parameters the 3000 application*/
/* is interested in receiving from the object manager */
opt_parms[0].keyword = OVc_OP_ATTR_LIST;
opt_parms[1].keyword = OVc_OP_LINKED_ID;
opt_parms[2].keyword = OVc_OP_ERRORS;
opt_parms[3].keyword = OVc_OP_OBJECT;
opt_parms[4].keyword = NULL;
OVcGetCnf(msg, &cntl, decode_buff, sizeof(decode_buff),        opt_parms);

/* extract values from attribute list returned */
attrlist = (OVcAttrList *)opt_parms[0].variable;
strncpy(NtwkInterfaceStarted,
        (char *)attrlist->v[0]->attributeValue.u.os.u.string.value,
        (int)attrlist->v[0]->attributeValue.u.os.u.string.length);
Flags = (int)attrlist->v[1]->attributeValue.u.os.u.number;
strncpy(NIProtocols, ..., ...);
MappingTableSize = (int)attrlist->v[3]->...
strncpy(MappingTableID, ..., ...);
   o
   o
   o

/* write data to dialog box using OSF/Motif function calls */
```

The Result!

After completing the 3000 object manager and application and the other object managers and applications defined in the system structure, we have an integrated network management solution for the HP Division LAN that looks like Figure 8. The network devices would actually show up as unique symbols. The main point is that the network shown in the original picture is now manageable. A user simply needs to know what object he wants to operate on and what type of operation he wants to perform. Each category of operations shows up as a menu on the HP OpenView Windows menu bar. The standard menus for operating on the network are: Monitor, Diagnose, Control, and Performance.

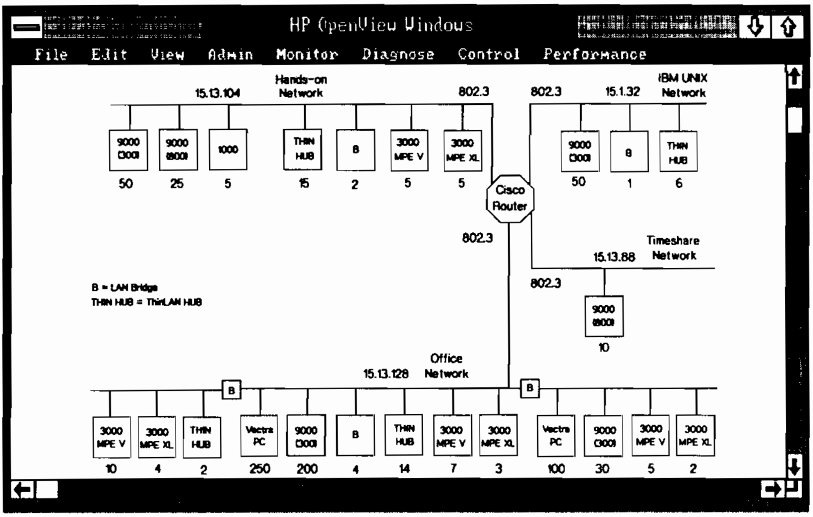


Figure 8. The IND LAN Solution

Network management solutions such as the one shown will decrease network downtime and increase network performance. The network administrators will enjoy a graphical representation of their network, a central place for all network events to be reported, and a common interface to all of their network management functionality. Problems will be detected more quickly and resolved more quickly so that companies can achieve the greatest utilization of one of their greatest resources--their network.

Understanding the Bridge and Router

John Vega
Hewlett-Packard
9606 Aero Drive
San Diego, California 92123

Introduction

The use of the "Local" Area Network, is quickly becoming the norm in most businesses. These independent departmental LANs are growing beyond the traditional LAN boundaries. This tendency is fueled by the availability of low-cost, digital transmission bandwidth. To prescribe the correct methodology required to accomplish the goals of the organization, the network manager must understand the functional differences between bridges and routers.

The Revelation

To understand the operation of bridges and routers, one must first gain familiarity with the way addressing is implemented within the data link and network layers. Every node on the network uses two addresses:

* Network Address (IP Address, at Layer 3) - used by routers
* A Device Address (Data Link Address, at Layer 2) - used by routers and bridges

Figure 1 shows the operational layers for the various devices that will provide for "Extended LAN" capabilities. Note that bridges are only concerned with layers 1 and 2. This means that bridges don't require the IP address, since it is in layer 3. Bridges operate on data link addresses. The data link addresses, one for source and one for destination, are known by several other names:

Data Link Addr. = Immediate Addr. = MAC Addr. = Station Addr. = Physical Addr.

Here is some background on why the data link addresses has so many different names. The data link address name is obvious, since that is the layer we're talking about. The immediate address is the "to and from" of the packet. The MAC address comes from the fact that the data link layer is composed of two sub-layers: Media Access Control (MAC) and Logical Link Control (LLC). Bridges operate at the MAC, or the lowest sub-layer. The station address is the address on the network interface card (physical address). You don't change the physical address. It's shipped with the interface and the number is controlled by the manufacturer. A block of numbers is given to each manufacturer. They allocate these numbers so that each interface has a unique address.

Standards Development

Various subcommittees of IEEE Project 802 produced the standards for layer 2. These standards are used for bridges. The American National Standards Institute (ANSI) produced the standards for layer 3. These standards are used for routers. The standards community, in an effort to avoid turf wars, purposely divided the responsibility for LAN-related OSI standards between IEEE and ANSI at the boundary between the data link layer and the network layer. Additional draft standards continue to be submitted by various working groups. The nice thing about standards is that there are so many to choose from.

Bridges

Local bridges directly attach network segments. Only a single bridge is required to connect two network segments. Remote bridges (also called half bridges) interconnect remote

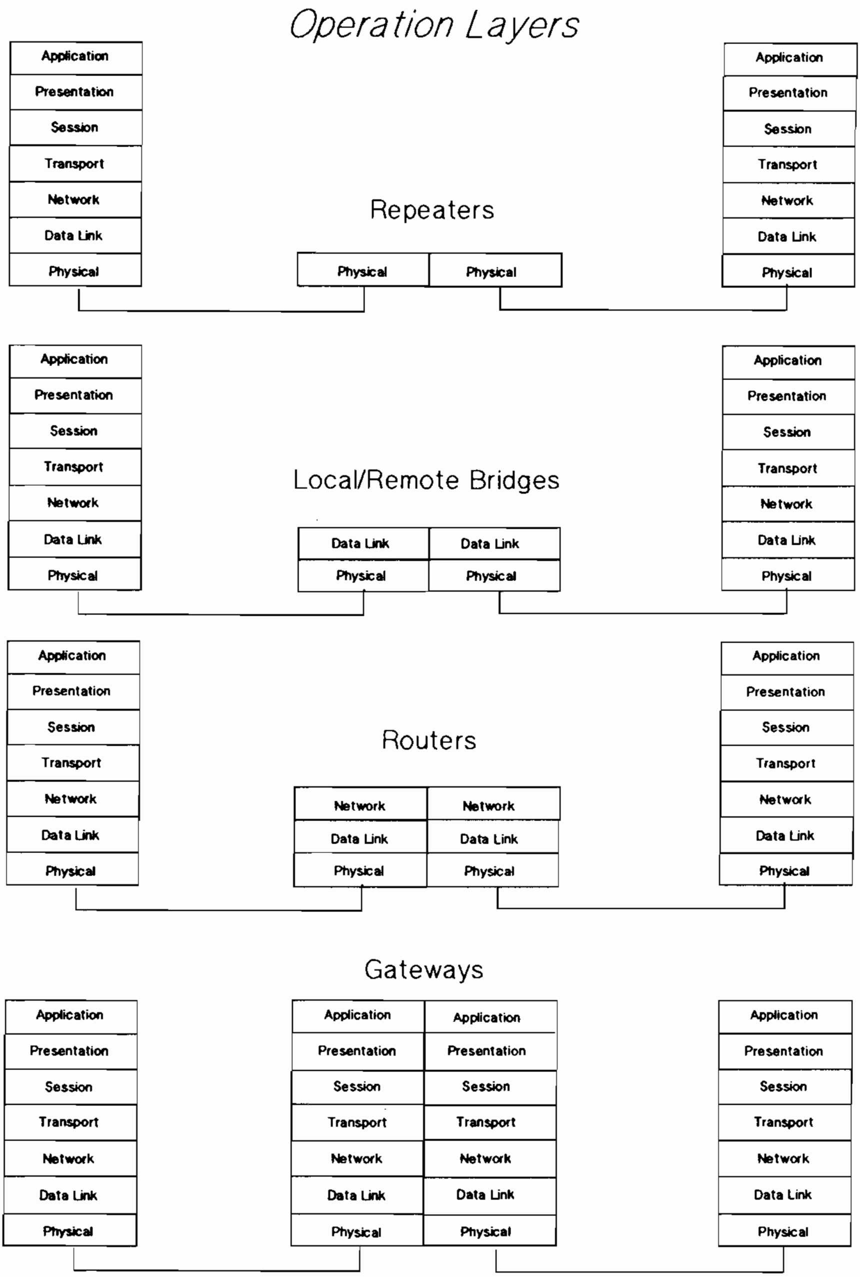

Operation Layers
Application
Presentation
Session
Transport
Network
Data Link
Physical
Repeaters
Physical
Physical
Application
Presentation
Session
Transport
Network
Data Link
Physical
Application
Presentation
Session
Transport
Network
Data Link
Physical
Local/Remote Bridges
Data Link
Physical
Data Link
Physical
Application
Presentation
Session
Transport
Network
Data Link
Physical
Application
Presentation
Session
Transport
Network
Data Link
Physical
Routers
Network
Data Link
Physical
Network
Data Link
Physical
Application
Presentation
Session
Transport
Network
Data Link
Physical
Gateways
Application
Presentation
Session
Transport
Network
Data Link
Physical
Application
Presentation
Session
Transport
Network
Data Link
Physical
Application
Presentation
Session
Transport
Network
Data Link
Physical
Application
Presentation
Session
Transport
Network
Data Link
Physical

Figure 1

network segments using some long haul transport mechanism (e.g. telco links). When connecting remote segments, two bridges are required. Bridges provide a store and forward function within layer 2, and utilize the MAC portion of this layer. This is why they are called "MAC layer" bridges. Processing addresses at the MAC layer can be carried out partly or entirely in hardware. Bridges are, therefore, very fast.

Bridges process *frames* and use layer 2 information to determine whether the frames should be passed from one network to another. This type of interconnection strategy is independent of the protocol used for OSI layer 3 and above. Due to the abundance of upper-layer (3-7) protocols, the independent characteristic of a layer 2 bridge is valuable. It means that any frame can be handled by a bridge as long as layers 1 and 2 conform exactly to the implemented standard in the bridge. The frame formats described in the MAC standard (*Figure 3, page 2.1*) include both destination and source address fields. The data link source address must always be the unique individual address of the transmitting station; the data link destination address can be either an individual or a group address (broadcast). Addresses at the data link layer are referenced as having a "flat" address space. This would be analogous to having a telephone system without an area code. The bridge provides a transparent access path across similar data link layers. Because of this transparency, no information can be sent to them since they are not individually addressed by nodes on the network. The bridge does have a data link address, it is used by Network Management software to manage the bridge.

When a Bridge is functioning, we will consider only the "Learning Bridge", it will filter and process all frames. In this manner it can build a data base of the "Data Link Source" addresses that are on its LAN segments. *Figure 2* shows a bridged network.

Bridged Network

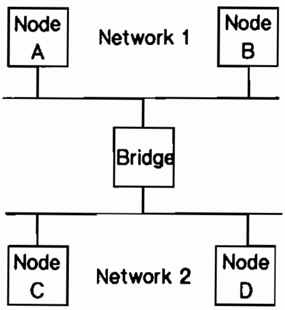
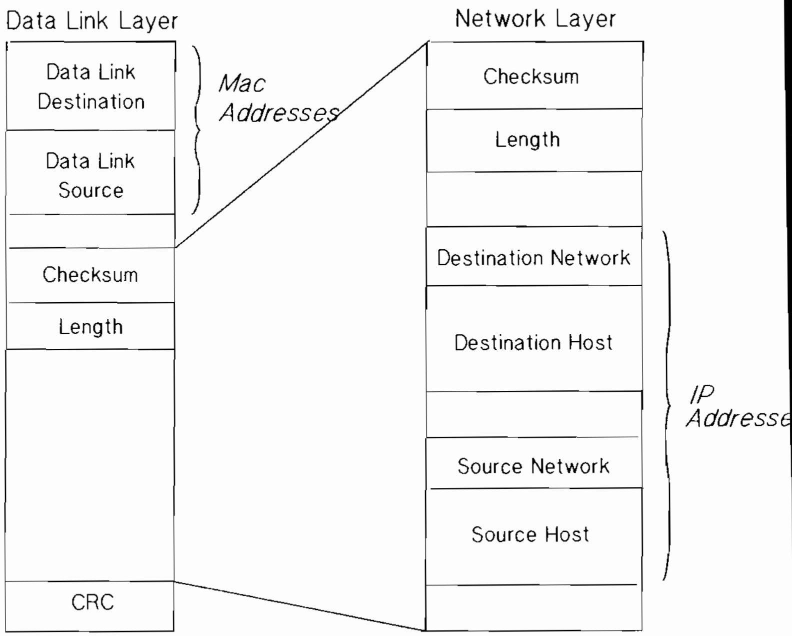


Figure 2

As the bridge "Learns" the nodes on the Network 1 segment, it can make decisions as to which frames to "forward" to Network 2. When the bridge is first powered up, and node A first communicates to node B, the bridge processes node A's request and copies the data link source address of node A into its data base. It then forwards the packet to Network 2. This action is commonly referred to as "flooding". Node B receives the original frame and sends a response to node A. The bridge processes node B's response destined for node A. The bridge knows node A is on Network 1 and does not forward the frame. Instead it copies node B's data link source address into its data base. The bridge now knows that both node A and node B are on the local segment. The bridge will discard all future frames sent between these nodes. In this way all source data link addresses are copied into the bridge's data base.

Addressing Overview



IP Addressing

Class	*Range*	*Example*	*Network*	*Host*
A	1-127	127.32.200.21	127	32.200.21
B	128-191	190.254.125.1	190.254	125.1
C	192-223	220.32.10.254	220.32.10	254

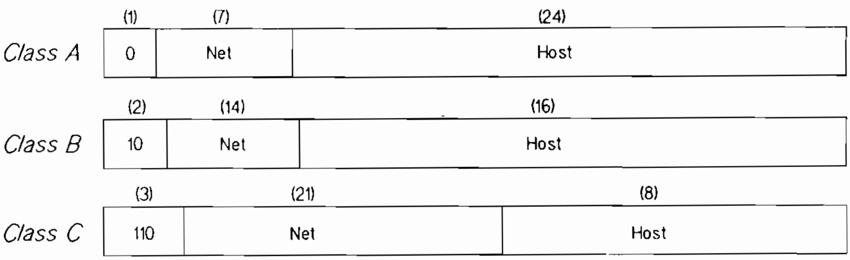


Figure 3

Forwarding takes place for any frame with a data link destination address not in the data base or for those frames whose entry, in the data base, indicates the node is on another network segment.

Bridges can have multiple interface ports. By filtering all source data link addresses on each port, a complete data base is built. Using this information, packet forwarding can be accomplished without "flooding" each port on the bridge. Automatic bridge maintenance controls the data base to eliminate old entries, "aging." The aging function removes addresses from the data base that the bridge has not heard from in a given period of time. Communication to a node that has been aged, would require the bridge to again flood the network to identify the node. Aging improves bridge performance, due to fewer items in the data base, and establishes a "true view" of the data link addresses on the network.

Reasons to Implement a Bridge

If the combined activity between all the nodes on a local area segment becomes extreme, network performance will degrade. If network traffic can be segmented, the resultant segments would experience fewer collisions and better performance would be realized. *Figure 4* shows how bridges can be used to segment a network. Once the bridge "Learns" the nodes on Network 1 and Network 2, it will forward only those packets that are not on the local segment. If node A and node C were PC Servers, users would typically do most of their work on a specific LAN server and be connected to the same LAN segment the server is on. Should they need to address the server on the other network, the bridge would forward frames to that server transparently. This approach would selectively isolate Network 1 from Network 2.

In many cases it is important to have an alternate path between networks. A learning bridge relies on a loop-free topology, only one path between networks. *Figure 4* shows a second bridge added to provide a redundant path between the networks.

Redundantly Bridged Network

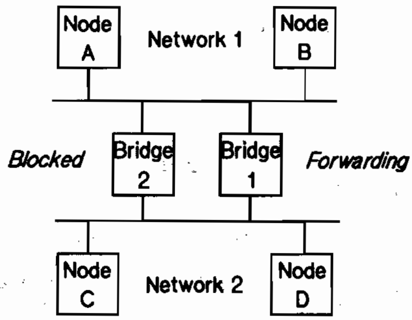


Figure 4

To maintain a loop-free topology one bridge must be "blocked," while the other bridge is in "forwarding" mode. If both were forwarding, and node A were to transmit a packet with an unknown destination (broadcast packet), both bridges would forward the packet. Each of the resulting copies would again be forwarded, and so on, until the entire bridged LAN would fail. The state of a bridge port is never changed from blocking to forwarding instantaneously. Intermediate states, called "listening" and "learning," are used to ensure no temporary loops are formed while the topology is stabilizing. The time involved for this switch is dependent on the size of the network. Creating this loop-free topology is termed "Spanning Tree", and means that there is only one path between LANs. The spanning tree algorithm and protocol is an IEEE Standard. The standard defines the protocol used between bridges. This protocol sets up the spanning tree and automatically provides for the re-mapping of the topology, should a bridge fail.

Router Revelation

Why do people call routers "gateways"? This "equality" is brought about by the naming conventions used for the routing protocols. These protocols are known generically as interior "gateway" protocols.

Why do we call IP networks that use routers "internets"? Because the protocol used for communication between routers depends on layer 3 "Internet" protocol.

Unlike bridges, a router can be directly addressed, at the data link layer, by nodes on the network. In a bridged network, the destination node is always addressed directly by its data link address. In a router network if the destination node is on the other side of the router, that destination node is addressed indirectly. The sender would use the data link address of the router instead of the destination node's data link. When routers are used, it is a requirement that the network layer address (IP address) be unique. Bridges do not care about the IP address.

Routers

Routers process *packets* and are protocol-specific devices. They operate using protocols implemented at the network layer. Some routers can run several network protocols at once. Among the well-established protocols are:

* (Internet Protocol), a part of TCP/IP, used extensively both commercially and in military networks

* IDS (Internet Datagram Protocol), a part of XNS from XEROX, used by many commercial vendors

* DECnet, Digital Equipment Corp's protocol

I will concentrate on the IP protocols for this discussion. TCP/IP uses internet protocol (IP) for Layer 3. *Figure 3, page 2.1* shows the format of the network layer. The IP address has two parts: a "Network Address," which identifies the network, and a "Host Address," which identifies a node within a network. There are three internet address classes, each accommodating a different number of network and host addresses. Their format is also shown in *Figure 3*. These addresses are distributed, by Department of Defense Advanced Research Projects Agency (DARPA), to suppliers of networking equipment. It is not necessary to comply with these addressing standards, although it is important to create addresses that have the IP format, when using the IP protocols. When a router is addressed, it looks at the network portion of the destination IP Address and if this address is different than the network portion of the source IP address, the packet is routed to its destination. This might be through another router, in which case the data link addresses, source and destination, are modified to access the next Router. It is the responsibility of the router to know the IP and data link addresses of the nodes the router is directly connected to.

Figure 5 shows a router network.

Router Network

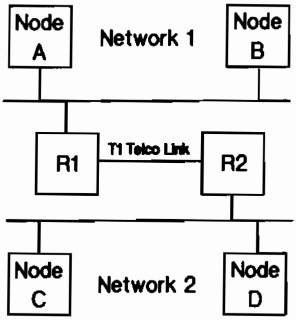


Figure 5

For node A to send a packet to node C, node A must manufacture a packet with node C's IP address and R1's data link address. R1 receives the packet and determines that the network portion of the IP Address references Network 2. R1 would then change the data link portion of the packet so that R1's data link address becomes the source and R2's data link address becomes the destination. When R2 receives the packet, R2 then changes the data link portion of the packet so that R2's data link address becomes the source and node C's data link address becomes the destination. The packet is then sent to Network 2.

Routing can be static or dynamic. With static routing, the paths to reach the different networks in the internet are configured in the system by the network manager. With dynamic routing, new paths are learned from the internet. Dynamic routing relies on standard router protocol to convey the routing information around the internet. When a topology change occurs, the routers directly involved in the change are responsible for initiating the reconfiguration. RIP (Routing Information Protocol) is one example of a standard routing protocol. RIP allows internet routers to build and maintain a routing table consisting of network addresses, route addresses and the number of hops to each specific network. Most vendors offer some proprietary extensions to RIP to allow their routers to make better use of the network's topology information. A second type of dynamic routing protocol, is based on the shortest path first. The algorithm is called OSPF (Open Shortest Path First), and is being proposed for use by the TCP/IP Community. It is being developed by a working group formed within the "Internet Engineering Task Force."

Routers work with what is essentially hearsay evidence, unless they are talking about themselves. This is sometimes called, "router gossip." The routing tables that they are constantly advertising and sharing are only as accurate as the last broadcast they received. A broadcast of a router's table might occur every 30 - 90 seconds, and whenever something changes.

An Analogy

One way to understand the internetworking abilities of routers is to make an analogy.

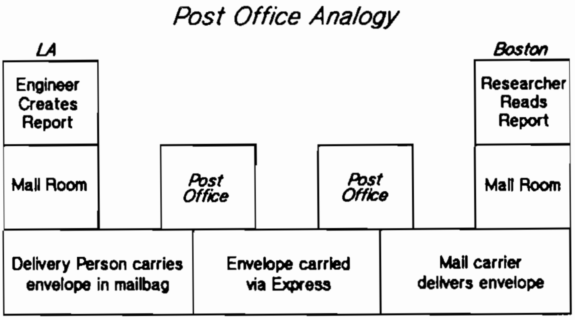


Figure 6

In *Figure 6*, Joe Engineer, in LA, wants to send a report to Mike Researcher, in Boston, using the US Postal Service. First Joe creates the report, places it in an envelope, addresses it and sends it to the mail room. The mail room puts the envelope in a mailbag and takes it to the post office. A postal worker then takes the envelope out of the mailbag, reads the address, and makes a decision on where to route the envelope. During the delivery, the envelope is forwarded by several intermediate post office facilities before reaching its destination. During each leg of the trip, the envelope is transported between post offices in a mailbag. Each mailbag has addressing information written on it that the postal service understands but is never seen by the original sender. Transportation between post offices can be by various means. These can include truck, plane, or train, whatever is most appropriate for efficient delivery to the next post office. At the final post office, the envelope is placed in a bag for a specific letter carrier, and is delivered to Mike Researcher's location. Neither the sender or receiver knows the path taken by the envelope.

Figure 7 shows how this analogy would relate to an internetwork environment.

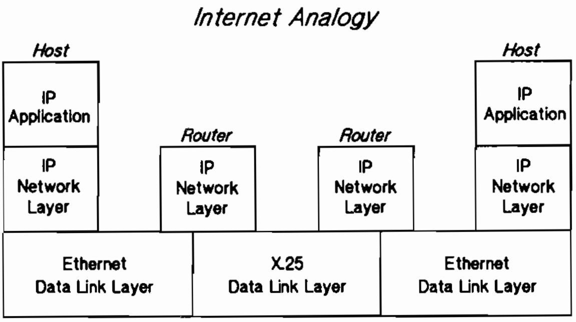


Figure 7

A Host computer sending information to another Host computer starts with a packet, formatted with an agreed-upon address. The Host transfers the packet to a communications interface that conveys the data through the network media to a router, equivalent to the post office. Just as the post office moves envelopes, routers receive data packets and forward them. A portion of the Zip Code is analogous to the network address within the IP Layer. Each Post Office knows how to get envelopes to any Zip Code, just like each router knows about how to reach each network in the internet. Only the final Post Office knows about the individual home or business addresses that it services. These addresses are analogous to the routers knowledge of the data link addresses assigned to each node on a specific segment of the network.

Using this analogy for bridges, would mean that the local bridge must be able to store every possible data link address a user would want to send a frame to. Where a router only needs to know every possible network address, or in our analogy, the network portion of the Zip Code. For example, if bridges were used to interconnect 200 LANs, each with 100 hosts, each bridge would need to store 20,000 data link addresses. This is because all LANs appear as a single LAN to the bridge. If a router interconnects the same 200 LANs, it needs to store only the 200 LAN network addresses.

Reasons to Implement Routers

Routers are protocol-specific devices, and operate using a network layer protocol. This implies that if the node you place on the LAN cannot operate under the routers Layer 3 protocol, you must bridge. This is true for HP's Datacommunications and Terminal Controller (DTC) and Digital Equipment's Terminal Servers. Each of these devices use proprietary layer 3 protocols.

When creating a network design, it is desirable to organize networks into reasonable size segments and provide isolation between the segments. The most important isolation is for control of multicast and broadcast packets. When connecting systems that use IP protocol, data link addresses are resolved by nodes in a specific way. Each Host must know who it wants to communicate with before it communicates. Identification of a node is by way of it's IP Address. To communicate to another node it is also necessary to have the data link address of the node you wish to communicate with. To resolve the data link address of a node, TCP/IP nodes use ARP.

ARP is an address resolution protocol designed to solve the problem of mapping between IP and data link addresses. When a sender needs to know the data link address of a receiving node, it broadcasts a packet that contains an ARP request that asks: What is the data link address for IP address X? The machine having the requested IP address sends back a reply that contains the desired data link address. The requesting machine then caches the response, and future mappings for that IP address occur quickly.

Probe is a special address resolution protocol used with Hewlett-Packard's proprietary Network Services. It works similar to ARP. One difference is that Probe uses Multicast rather than Broadcast packets. The Multicast packet is more selective, in that it will only be acknowledged by systems that are running HP's Network Service protocol.

Each system knows its own IP and data link address. If the sending node is to communicate to another node it must resolve the data link address of the receiving node. The effects of this resolution highlight an important difference between routers and bridges. If a bridge processes an ARP broadcast, it must "flood" the ARP broadcast packet through the network. This is done whenever a broadcast is sent, by any node on the network. The router, by definition, will block all broadcast and multicast packets from being forwarded over the network.

In *Figure 8*, node A sends a broadcast to the network, the router decodes the destination network address and concludes that it is the same as the source network address, and blocks the broadcast from being sent across the network.

ARP Blocking Router

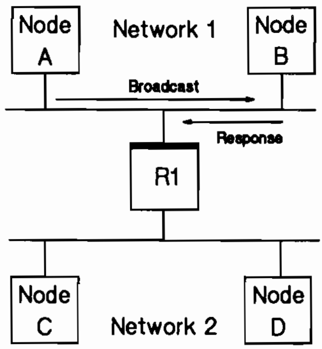


Figure 8

ARP Forwarding Router

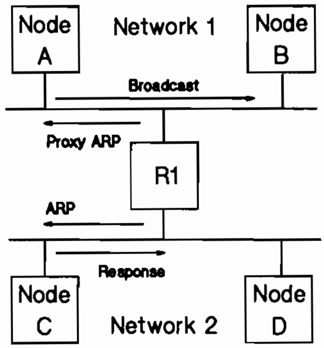


Figure 9

In *Figure 9*, node A sends a broadcast to the network. The router decodes the destination network address and concludes that it must route the packet. The router responds to the ARP broadcast, with its own data link address. This response is known as a "Proxy ARP." This satisfies node A's request, and from this point forward, node A will use the router's data link address to get to node C. The router sends the packet to Network 2, using the routers source data link address along with node C's IP address. If the router knows the data link address associated with node C's IP address, it sends the packet to node C, using the router's data link source address and node C's data link destination address. In this way the router tells node C that if further communication is necessary, it is to use the router's data link address when communicating to node A. If node C's data link address is not in the router's cache, the router issues an ARP of its own. Node C responds, and node C's IP and data link addresses are placed into the router's cache for future use. The packet is then sent to node C, using the router's data link source address and node C's data link destination address.

As larger networks are constructed, it is essential to have some form of dynamic routing. In a network of bridges, with redundant links, only a single active path can exist between network segments (spanning tree). In a router network, the router can choose the optimal path from among all routes available. It can also use additional sophistication such as: congestion control, type of service (delay, throughput and reliability) and packet fragmentation. The router is a true "Network Citizen."

Traffic Considerations

The most common limitation in bridges and routers are memory and communications interface capacity. With bridges, there must be enough speed to filter and process every frame on all attached interface ports. This means it is possible for a bridge to run out of processing capacity even when no traffic is passing through it. When this happens the bridge "discards packets." In addition, the bridge must decide which port the frame should be forwarded to. A router needs to filter all frames on its interface and process those frames that directly address the routers data link address. The router also processes broadcasts when the destination network address is different than the source network address.

A 10-Mbps Ethernet can carry minimum-size (64 byte) back-to-back packets at a maximum theoretical rate of 14,880 packets/second. For remote networking, a T1 (1.544Mbps) channel can carry 3,000 packets of 64 bytes each. If the link were a 56Kbps, it could carry 190 packets of 64 bytes each. When bridging/routing between any two segments, it is important that the bridge or router filter very quickly. The forwarding rate is limited by the available bandwidth connected to a particular port. The forwarding rate of a bridge will be much higher than that of a router. When comparing filtering and forwarding rates, it is important to understand if the performance numbers indicate a single interface card with multiple ports or multiple interface cards.

Conclusion

The properties of bridges and routers differ. Each has its own unique advantages and disadvantages. The decision whether a router or a bridge is the better choice to connect a specific pair of data links depends upon the environment.

Because of the low level they operate on, bridges are generally more susceptible than routers to being overloaded as a result of unexpected traffic surges. Most bridges will perform best in a simple single-link LAN-to-LAN configuration, especially over shorter LAN-to-LAN distances and where forwarding decisions are few and straightforward. As network complexity grows, and where the LAN-to-LAN traffic traverses mesh or loop topologies, routers are generally more suitable. In general, routers are clearly preferred, where LAN-to-LAN traffic involves three or more remotely connected LANs, multiple T1 links or multiple high-level protocol stacks that operate concurrently over the interconnected LANs.

References

IEEE Network: January 1988 - Volume 2, No.1

Unix Networking: Stephen G. Kochan and Patrick H. Wood

Data Communications: Special Issue - Sept. 1989

UNDERSTANDING BASIC CABLING: STARLAN AND SITE-WIRE

By Charles H. Wirl, President
Wilco Communications Inc.
P.O.Box 60579
Sunnyvale, Ca. 94088
Ph: (408) 746-3993

Success in the data center, as in many other areas of business, depends on making the right connections. In this case it means making the right connections between your computer and its many terminals.

For network managers, getting "wired" involves much more than to little sleep or to much caffeine. As telecommunication systems have increased in both complexity and number, so too have the requirements for complex and complete building wiring systems. Gone are the days when wiring systems were "voice only" and data devices simply required point-to-point links.

Today's communications managers are seeking wiring systems that satisfy network needs, including total support for multiple vendor voice and data systems, one time installation of wiring components, ease of equipment moves and changes, and simple cable management.

SIMPLE CABLING SCHEMES

* Simple RS232 and RS442
* ADCC and ATP type connectors
* Use of Octopus Cables and Patch Panels
* Use of 66 blocks
* Testing of total modular system

CABLE MANAGEMENT

A cable management system should be set up and constantly updated when adds, moves or changes are done to the system. Documentation is the key to keeping on top of the mounds of cable.

Documentation

Every cable must be labelled with the same number at both ends. When installing wall plates, also label the cable inside the wall plate. This will assure that the identification number will not be lost, even if the wall plate is replaced.

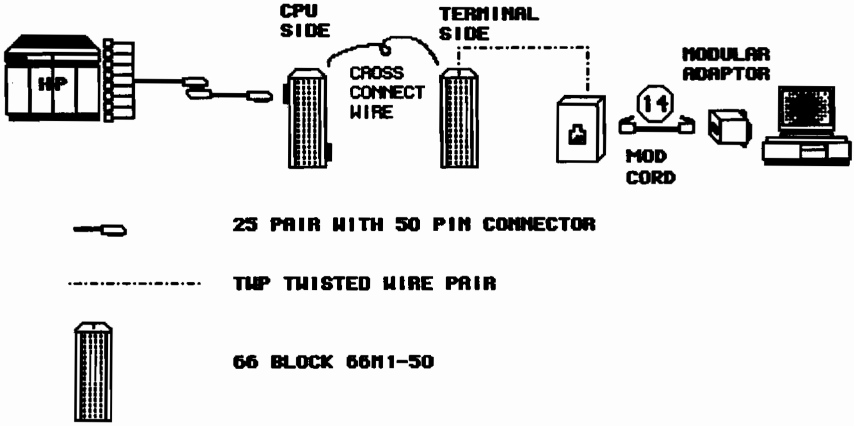
When wiring the building and especially when adding new runs, always use the same color code for wires. Try to follow in someone's footsteps, even if what they had done is not what you would normally do. It will save time in the long run.

EQUIPMENT

A 66-block (aka RJ-21x, M1-50, punch down block) is a wonderful invention. It allows you to connect (splice) a twenty five pair cable to individual workstations. There are two hundred little silver pegs on each block. Each block handles two cables, one on the left and one on the right. One set of fifty pegs is used to punch down telephone cable #1. The next set receives the terminal cables, thereby splicing them to telephone cable #1. The process is repeated for the second telephone cable. The 66 blocks are also made with a connectorized socket so you plug in the telephone 25 pair cables directly and just punch down the terminal wires.

Another feature of a punch down block, is that it allows you to make a cheap patch bay. You would have two sets of blocks, the left set comprising all of the ports from the computer, and the right set of blocks connecting to the terminals. Then by cross connecting the left and the right blocks, you can activate selected terminals. Figure 1 shows the punch down blocks used to cross connect terminals.

FIGURE 1



MODULAR CABLING

The modular telephone adapter is now being used for many different computers. For the Hewlett Packard HP/3000, the RJ11 four wire jack is used. You run your standard two pair cable to an office and then install a modular RJ11 wall plate. A flat modular telephone cord runs from the wall plate to the terminal. Unlike that for the telephone, it is pinned straight through.
An RS-232 to RJ11 converter is attached to the back of the terminal, which converts to modular cabling.

Whenever you vacate an office or cubicle, you would remove the modular cord. This makes the cable less prone to getting run over by the cleaning lady's vacuum or the moving-man's dolly.
The idea of modularity also allows you to easily change the existing configuration. At the terminal end dumb terminals and pc computers use different connectors, the modular concept allows the user to change the mod adaptor instead of changing the connector on the end of the cable. At the CPU end the user has the choice of using any port to any terminal by just plugging and replugging a telephone modular cord.

HEWLETT-PACKARD SiteWire

HP SiteWire is Hewlett-Packard's comprehensive mulitvendor premises wiring program for both voice and data applications.

User Area

The user area (work area) consists of the telephone, personal computer, and perhaps other devices such as a printer. The computer cable feeder plugs into the workstation and into the walljack. It has an 8-pin plug on both ends. The workstation can be supplied with an RS-232 to 8-pin adaptor.

Horizontal Area

The horizontal area consists of the wiring closet, the wall plugs in the user area, and the cabling between the two.

Vertical/Backbone Area

The vertical portion interconnects the various wiring closets and the equipment rooms for PBX and computer data centers.

Equipment Room

The equipment rooms normally house the PBX and data switch equipment, and computer data center equipment.

Administration Areas

The administration areas include the wiring closets and the main equipment room. The PDS twisted pair wiring provides an easy method of wiring administration.

HP StarLAN

The StarLAN product is based on the IEEE 802.3 technology. It provides a link level protocol that insures complete integrity of data, and a reliable communications network. It uses the Twisted Pair wiring to provide high reliability, high flexibility in administration, low initial cost and administration over the life time of the wiring plant. The StarLAN design utilizes one megabit per second transfer rates, with a Bridge to the IEEE 802.3 network.

Each StarLAN Hub provides 12 jacks for the support of eleven PCs plus a connection to another Hub. Network design recommendations suggest a limit of 50 PCs per StarLAN subnetwork. This is based on typical traffic patterns in Office Automation workgroups. The StarLAN support over PDS twisted pair wiring will support distances of 250 meters between PC and Hub. This is more than adequate for any possible configuration on the floors of most buildings.

StarLAN 10 Hub

The StarLAN 10 Hub is a 10 Mbps multiport repeater designed for IEEE 802.3 and Ethernet local area networks operating over unshielded twisted-pair cable. This hub supports twelve unshielded twisted-pair connections and distances up to 100 meters between the hub and the computer or DTE.

The HP StarLAN 10 Hub connects HP StarLAN 10 horizontal subsystems to baseband coaxial backbones.

CAMPUS ENVIRONMENT

A campus environment takes more than one complex building and makes an overall cabling system very manageable. Each building is separated into a floor with at least one main wiring closet. All terminals on that floor run to that wiring closet. From that closet, the runs are punched down to 66 blocks (devices to splice cable). Twelve cable runs are channeled into one 25 pair cable.

These cables are called the backbone wiring. These runs and all other floors having terminal cables cross-connected to a backbone run meeting in a main building facility closet. If the computer room is in this building then the cables would be bridged to the computer room. A campus backbone would then run from this building to the next building.

Within each floor it is possible to sub-divide locations into other cabling closets. The standard 2 pair cable is run from the terminal to this closet. From the closet a twenty five pair cable would run to the main floor wiring closet or to the computer room wiring closet.

Each terminal cable should be labelled and identified. Remember to label the 25 pair cables so you easily identify both the *floor* and *exact location*. Telephone lines and data lines are usually found in the same wiring closet. Data communications lines such as leased lines for remote sites are easier to run to the computer room and it is also easier to diagnose problems when both lines are in the same location..

TESTING CABLES

You may want to test a cable to make sure it hasn't been broken at some point in the install. You will need an ohmmeter and a piece of wire. Go to one end of your run and "short out" pins 2 and 3 by strapping the two pins together with the wire. Then go to the other end and put the ohmmeter on pins 2 and 3. The ohmmeter should read zero, indicating a shorted position. Next, check pins 7 and 20 in the same manner for continuity. This test will prove whether or not the cable has been broken or shorted. The ohmmeter should read open (or one for digital ohmmeters) when all jumpers are taken off. If not, the cable may be shorted somewhere along the run.

A more expensive device used to diagnose problems is a Time Domain Reflectometer. It sends a waveform of a certain frequency down the cable. By the amount of elapsed time it takes for the wave to bounce back, it calculates how many feet to a problem area. These may be shorts, opens or even splices. A TDR is especially handy when troubleshooting coaxial Local Area Network problems It can also measure the distance of each cable run.

A toning device is used by the telephone company to identify cables. They usually are yellow and make a siren-like sound. Toning sets are used to trace cables that were not properly labelled, missing,or lost. The wand is then waved over a mound of unlabeled cables and when the right one is found the wand makes the paramedic noise.

CONCLUSIONS

The many advancements in network and workstation technology have made it more difficult to design a network scheme and select components. Managers need to remember that the system is only as good as its weakest link. Networks are complex and change comes about too fast for most managers to keep up with.
The key to reducing the impact of changes in today's complex world is through cabling flexibility and modularity.

ARPA Services for MPE XL

by Deborah Campbell
Hewlett-Packard Company
19420 Homestead Road
Cupertino, California

I. ARPA Services for MPE XL - Product Implementations

A. Introduction

Hewlett-Packard is providing the Telnet (Teletype Network Protocol) and FTP (File Transfer Protocol) ARPA Services for MPE XL. These services run over a TCP/IP (Transmission Control Protocol/Internet Protocol) transport stack. This transport in turn runs over Ethernet or X.25 networks.

Because many vendors are providing the same services, transport stack and network connections on their systems, the MPE XL Telnet and FTP services allow the MPE XL system to share information over networks with these other vendors' systems.

B. HP's FTP and Ethernet Implementation

FTP has been implemented as a software product on the MPE XL host. Every MPE XL system requiring FTP communication will need a copy of this FTP software product. Ethernet has been implemented as part of the ThinLAN 3000/XL Link product. For customers who already have the ThinLAN product, Ethernet is provided as a software upgrade. No additional hardware is necessary.

FTP will run over the ThinLAN and DTC X.25 3000/XL Network Link products. FTP and NS can simultaneously use the same ThinLAN Link, with NS running over 802.3 and FTP over Ethernet. FTP and NS can also simultaneously share the same DTC X.25 Link.

C. HP's Telnet Implementation

Telnet for MPE XL is implemented on the DTC (Datacommunications and Terminal Controller). By choosing this implementation instead of a system-based implementation, Hewlett-Packard has created a Telnet front-end processor for MPE XL. Why implement Telnet differently than other vendors? Why not put it on the host? Because Hewlett-Packard wanted to provide the best solution for MPE XL.

1. Some background on MPE XL and the Telnet protocol

The HP 3000 with its MPE XL operating system was designed and has been tuned for OLTP (On-Line Transaction Processing), the input and retrieval of data through applications. Therefore, the system is most efficient and productive when manipulating large blocks of data, typical for OLTP applications. VPLUS and HP Blockmode terminals are examples of specialized HP 3000 tools that Hewlett-Packard has developed for the OLTP application environment.

Computer Museum

Telnet is a character-mode networking protocol. Each packet sent on the network contains only one character of data. Telnet therefore creates a large amount of processing overhead on any system.

On MPE XL the impact of this type of processing is extremely high, again because MPE XL is not designed or tuned for character-oriented transactions. (UNIX[1] and VMS[1] [2] are character-oriented operating systems, but are therefore not as good for OLTP. It's an efficiency tradeoff.) We saw the need for multivendor access to and from the MPE XL systems via Telnet as a critical and significant one for our customers, and determined that the best solution would be to offload the MPE XL system of the Telnet processing. By choosing to implement Telnet this way, we reap the following benefits:

- offload the MPE XL CPU, which in turn
    - provides more CPU for applications
    - allows support of more users
    - provides higher performance Telnet sessions
    - achieves cost-effective modularity by allowing additional users to be added through an additional front-end solution instead of an additional system
- provide a single solution for multiple MPE XL systems

We took full advantage of the DTC's ability to take character-mode input and optimize it for OLTP, and created a solution that provides significantly better cost- and performance-per-session benefits than a system-based solution ever could.

2. Implementation Details

An MPE XL user on a terminal (or PC) connected via a single port to the DTC accesses a remote Telnet host by using Telnet directly to that host, without going through the MPE XL system. (We refer to this functionality as "outbound Telnet access".) This same user connects to the MPE XL system from the same terminal using HP's high performance terminal connect protocol AFCP (Advanced Flow Control Protocol). Therefore, a single user can connect to several systems, HP or non-HP, from the same terminal on the DTC. This DTC can be a DTC 48 or a DTC 16.

A user on a remote Telnet host accesses the MPE XL system via two possible solutions, a new card for the DTC 48 or a dedicated Telnet front-end solution. (We refer to the functionality provided by both of these solutions as "inbound Telnet access".)

Both inbound Telnet access solutions provide

1 UNIX is a registered trademark of AT&T in the U.S.A. and in other countries.

2 VAX is a registered trademark for Digital Equipment Corporation.

- Telnet-to-AFCP protocol conversion, optimizing Telnet for the MPE XL environment.

- the ability to access multiple MPE XL systems with a single solution.

- the option of making the solution transparent to the end-user.

The two inbound Telnet access solutions have some differences. The dedicated Telnet front-end solution supports many more users than the card for the DTC 48. The new Telnet card for the DTC 48 can be used in conjunction with Asynchronous Connector cards to provide inbound and outbound Telnet access, and MPE XL terminal access in the same DTC 48.

On first release

- only one inbound Telnet access card is supported per DTC 48.

- X.25 and inbound Telnet access are not supported in the same DTC 48.

- the inbound Telnet access card is not supported in the DTC 16.

- all Telnet solutions require the OpenView DTC manager.

Telnet (inbound and outbound) access is supported over Ethernet and X.25 networks. Telnet will not be supported over the DTC X.25 3000/XL Network Link, but will be supported over routers to X.25 networks. When using X.25, PAD is preferable to Telnet as an interactive access method.

Telnet, as a character mode protocol, is expensive over X.25. The heavy traffic can increase the line cost and reduce performance. PAD is tuned for interactive access over X.25. PAD provides interactive connectivity at a lower line cost and higher performance level than Telnet.

Customers who need inbound and outbound Telnet access for a limited number of users would use the DTC 48 solution.

Customers who need only inbound Telnet access would use the dedicated Telnet front-end solution.

Customers who need lots of inbound Telnet sessions and only a few outbound sessions, or who have the DTC X.25 Network Link would use the DTC 48 for outbound Telnet, terminal access and X.25 connections along with the dedicated front-end solution for inbound Telnet access.

II. Supported Configurations of ARPA Services for MPE XL

(Please see the product data sheets for the specific hardware, operating system and ARPA software versions.)

On first release both Telnet and FTP will be supported with the following remote systems:

- HP 9000/300
- HP 9000/800
- VAX/VMS with WIN/VX
- SUN workstation
- Vectra with ARPA Services 2.0

Telnet will additionally be supported with:

- DTC

FTP will additionally be supported with:

- Apollo workstation
- HP 1000
- HP 3000/MPE V with WIN/TCP for MPE V
- HP 3000/MPE XL

III. HP's System-to-System Network Strategy

HP's "system-to-system" network strategy is the strategy for interconnecting multiuser systems.

Hewlett-Packard is committed to international and de facto industry networking standards. Towards that end, we are providing ARPA and OSI services on all of our computer systems. Additionally, for connectivity between HP 3000s, we offer NS3000 Network Services.

Customers who need the broader set of services for the HP 3000 would use NS3000 Network Services now, migrating to OSI in the future.

Customers who need multivendor communication would use the ARPA services now, migrating to OSI in the future.

TITLE: Migrating to NS - How and Why

Author: Ann Smith

AUTHOR: For more information contact:

Ella Washington

Hewlett-Packard

19091 Pruneridge Ave. MS 46LK

Cupertino, CA 95014 408-447-1053

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 7233

Managing HP3000 Performance

Brian Nogar & Chris Hauck
Hewlett-Packard Company

Executive Summary

Maintaining superior system performance is a much sought after, but often times elusive goal. The dilemma which frequently arises is that once an acceptable level of performance has been achieved, increases in system workload occur which consume the excess capacity. New applications, more users, business growth, corporate mergers and so forth are all examples of changes which affect the way computers perform.

In order to adapt to these changes while maintaining an acceptable level of performance, some form of performance management must be practiced. This paper approaches performance from a management perspective and outlines a specific process to use for managing performance. The concepts behind the process, how to design and implement the process and performance management best practices are the main topics discussed below.

Before delving into the details of the Performance Management process, the next section will spend a little time addressing why Performance Management is important.

Why Manage Performance?

Performance Management is the prediction and avoidance of performance problems, while performance troubleshooting is the detection and correction of performance problems. It is much less time consuming to avoid problem situations as opposed to encountering problems and spending the time required to solve them.

Unfortunately, performance needs historically have been addressed on a problem solving basis. Systems would continue to grow and evolve until performance reached unacceptable levels, and at this point performance issues would begin to be addressed. This approach usually led to intensive troubleshooting efforts, as well as rushed equipment orders. Because of the time constrained and pressure packed nature of the troubleshooting environment, the most efficient and cost effective solutions were often overlooked.

In addition to the troubleshooting required, this method of managing performance has a number of other significant drawbacks. Perhaps the largest drawback, is that a very reactive environment is established. Constantly responding to system needs in a reactive manner causes the system manager to lose control of the management activities which need to take place. In essence, the system becomes the controlling presence in the environment. This creates a very unproductive use of both time and resources as time spent addressing unexpected conditions diminishes the time available for productive management tasks.

The benefits found in the performance management approach are numerous. The entire focus of the performance management process is a proactive management style rather than a reactive style. Time is allocated up front to be spent performing tasks designed to improve system utilization. An example of this type of activity is the development of a list of performance maintenance tasks. As a part of the performance management plan a list of such tasks coupled with an implementation schedule is compiled and documented. The types of tasks found on this list include activities such as: database maintenance(Master capacity, detail re-packing), disc space management, application analysis, monitoring schedules, etc.. These tasks are all designed to have a positive impact on overall system performance. By automating the execution of these tasks overall system throughput can be improved without a high labor overhead.

The performance management process greatly increases the ability to align system needs with business needs. By managing performance and anticipating growth it is much easier to forecast future budget requirements. This enables system growth to become closely aligned with business growth and facilitates the entire system planning process.

Performance Management - Concepts

Managing system performance is a task that everyone performs to varying degrees. Setting limits, managing disk space, performing data base maintenance, etc. are all examples of typical tasks utilized to improve performance. Performance Management is nothing more than formalizing these types of procedures and using a well defined methodology to monitor and analyze performance in an on going fashion.

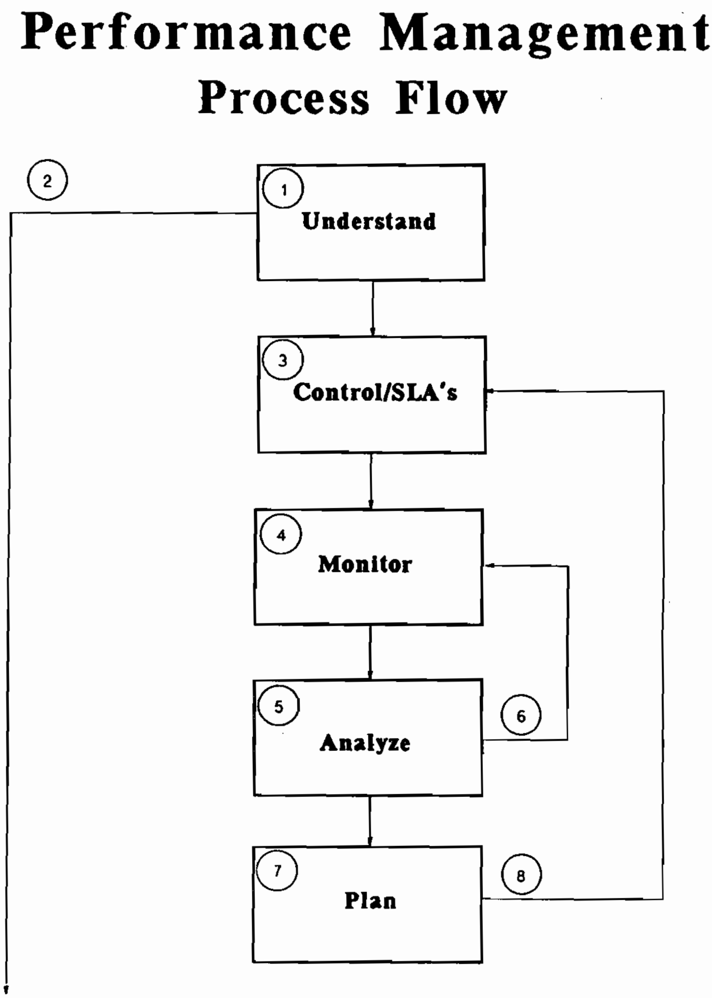
Outlined in figure 1 are the five steps which make up the Performance Management process. Details for each as well as how they relate to each other in forming a cohesive process will be provided throughout the report.

Figure 1

 **Understand**	Before implementing a performance management program a base level of information is required. Performance analysis concepts, system goals and expectations, and the current system utilization profile are all areas which should be researched before a performance management program is implemented.
 **Control**	Control involves setting the performance guidelines which your organization will adhere to. Guidelines are developed in the areas of acceptable performance levels, measurement techniques, performance maintenance procedures, and planning methodologies.
 **Monitor**	Monitoring is the on going process of measuring system performance levels. A well defined and consistently administered monitoring schedule provides a periodic check-up (or measurement) of system utilization.
 **Analyze**	Performance analysis is performed to insure that the system goals and requirements are being met. Analysis is the method of detecting any bottlenecks to system performance that may be developing.
 **Plan**	Planning is required whenever analysis determines that a potential bottleneck is developing. Planning involves identifying all available solutions, and then determining which one is the most effective from both a performance and a cost perspective.

Each of these stages all add a distinct piece to the performance management puzzle. Together they form the integrated process which is used to manage performance in an ongoing fashion. Figure 2 provides a flow chart detailing how each phase in the process interacts with the other stages.

Figure 2



The legend below outlines the steps taken in implementing the performance management process. The step numbers below correspond to the labeled sections on figure 2.

1. The process begins with information gathering. Research on such topics as current system utilization levels, system goals and expectations and basic performance analysis concepts is required to start the process.

2. Information gathering is a never ending process. As long as the system is being managed, new information will be gathered and applied to the process.

3. Defining the control parameter framework is the building block of the process. This step involves defining the minimum service levels the system should maintain, defining system maintenance tasks to be regularly executed, and creating a schedule for the entire process.

4. Monitoring is performed to gather information concerning how the system is performing. Monitoring must be performed on a regular basis to insure no service levels are being approached or exceeded.

5. Analysis is performed to insure the system is reaching its' potential. Analysis involves comparing the measured results against the service levels as well as ensuring no performance bottlenecks are developing.

6. If analysis reveals that no service levels are being approached and no bottlenecks are developing, then no action is taken. The next step in the process is to execute the next measurement interval.

7. If a potential problem has been detected, planning is utilized to determine the best alternative to alleviate the problem. Planning could involve additional hardware, application changes, workload re-distribution or re-setting the service level expectations.

8. Once a planning alternative has been implemented - the control parameters must be re-evaluated to see if any changes are required.

From the description above, it is obvious that the process is a fairly simple one to implement. However, management commitment is required for it to be successful. This commitment is necessary in the establishment of service level agreements as well as committing the resources required to monitor and analyze regularly.

A key point about the process is that it should evolve to reflect the changing business needs. The exact format and structure is very flexible and can be customized to meet the needs of every account.

Performance Management - Service Level Agreements

Before discussing the specifics of how to develop and implement the performance management process, it is necessary to spend a little time discussing the topic of "Service Level Agreements"(SLA's). Although the concept of service level agreements has been in existence for a long time, SLA's are relatively new to the HP3000 world. An understanding of service level agreements is very important because they are integral in the creation of the performance management process.

A service level agreement is essentially a contract between MIS and the user community which specifies that MIS will deliver a minimum defined level of service to the users. SLA's are mutually developed between MIS and the user community, with MIS soliciting input from the users concerning system expectations, and the users providing MIS with their expected utilization and growth levels. At this point, an agreement is created which defines the expectations of both sides. This agreement will not only specify the minimum level of service for such areas as response time, throughput, and system availability, but it will also define the system metrics used to verify service level achievement.

Benefits abound for both sides in this type of agreement. From the user perspective it is very simple, they have an agreement which documents the specific level of service (response time, system availability, etc.) which the system is expected to deliver. An added bonus in this arrangement is the fact that the users were actively involved in determining the appropriate levels.

For MIS, the benefits might not be quite as obvious, but they are probably more significant. The most important benefit for MIS is that a Service Level Agreement establishes the baseline system performance levels that must be met. Once a baseline has been established, monitoring, analyzing, and planning methodologies can be developed to insure SLA achievement. An additional benefit for MIS that SLA's offer is that the expected user utilization and growth needs are also documented in the agreement. This simplifies the planning process tremendously.

With the introduction of SLA's under our belt, it is now time to look at the steps necessary in developing the process. As a review, the five steps in the process are: Understand, Control, Monitor, Analyze, and Plan. Each individual phase will now be examined in more detail in order to discuss the alternatives available.

Understand (Information Gathering)

The first step towards creating the plan is to establish a base knowledge level. Background information should be obtained in three basic areas: basic performance concepts, current system environment, and the business/system objectives.

To facilitate the learning process a brief overview of HP3000 system performance terminology and concepts will be discussed next.

Performance Basics

How do you define good system performance? If you were to ask 10 different users this question you would likely receive 10 different answers. The way they use the system, the application being run, their job requirements, etc. are all factors which form their performance expectations. Factor in individuality and it is easy to understand why everyone provides a unique response.

As we begin to manage performance, the ability to quantify system goals, expectations, and utilization levels is required. This requirement is for some method of measuring performance. The three basic measurement methods used today are: Throughput, Response Time, and Resource Utilization.

Throughput is nothing more than a measure of how much work the system is accomplishing. This measure can take place using multiple metrics. Typical throughput measures include:

Transactions/Hour
Jobs/Day
Orders/Hour(Day)

These rates provide a quantitative measure of how much work the system is executing in a given time period. Transaction and jobs rates can be counted by the system(using a tool such as LaserRx or HP Glance) or an application can be designed to collect logical or application specific transaction rates. Application rates, such as orders per day, are very good measures for aligning system usage with business requirements.

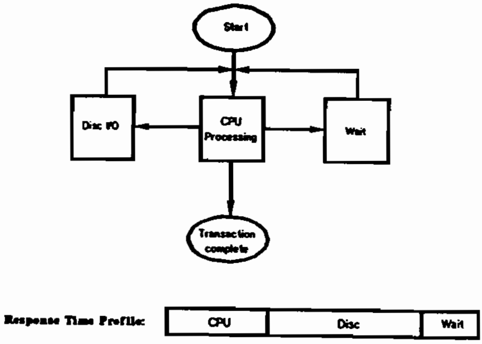
Throughput is also a very good method of measuring system growth. By comparing throughput rates from different time periods a comparison can easily be made concerning how much growth the application has sustained.

Response time is the amount of time a user waits for the system to process a request and respond with the proper output. The determination of acceptable response time will vary from user to user depending on the application and user expectations.

Since response time is the primary method most users utilize to measure performance, determining what systems factor affect response time is very important. Figure 3 provides a schematic of what functions the system performs while responding to a request.

Figure 3

Response Time



This figure illustrates that the time spent waiting for the system to process a transaction is broken up into several distinct components. The three most common components are : CPU time, Disc I/O time, and Wait time. The flowchart illustrates how time is accumulated in each component. As a minimum every transactions accumulates some amount of CPU time. If the transaction requires Disc I/O, the process will give up the CPU while waiting for the DISC I/O to complete. Once the Disc I/O completes and the process is rescheduled, CPU time begins to be accumulated as the programs' instructions are executed. The process will stay in the CPU phase until one of the following actions occur:

- The transaction completes
- Another Disc I/O is requested
- The Process waits for another reason(File/Data Base lock, non Disc I/O, etc.)

The total time spent in each of the components is accumulated and creates what is known as the Response Time Profile. The response time profile is very important when analyzing application performance as it provides a picture of what the limiting resource to performance is.

If the response time profile looks like this,

it is an indication of a CPU bound application. The best alternative for improving performance is to either perform some application optimization(using APS or SPT) or upgrade the CPU.

On the other hand, if your response time profile is dominated by Disc I/O time,

CPU	Disc	Wait

then any troubleshooting time would best be spend analyzing the data base and I/O subsystem design.

A response time profile which shows Wait Time as the major component will require more analysis to determine what the application is waiting for. Although a typical reason is a poorly designed locking strategy.

In general the response time profile is very important in managing performance. As shown above it can be very helpful in a troubleshooting environment, by highlighting the areas which need the most attention. It can also be useful in the planning area, by providing a firm understanding of how the application works as well as the typical application resource requirements. Response time profile information is typically acquired by using a performance management tool. Both HP GLance via the STATE bar on the PROCESS menu, and LaserRx in the DETAIL ZOOM of APPLICATION data, provide response time profile information.

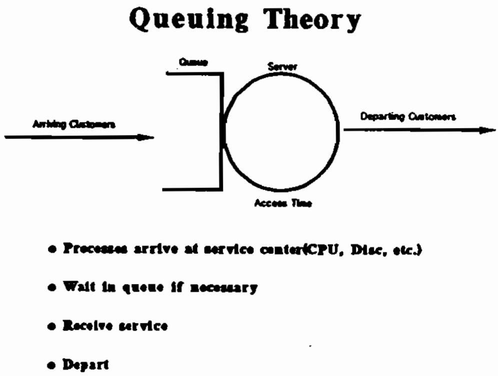
The third method of measuring performance is Resource Utilization. A computer system can be thought of as a collection of "service stations" working together to execute integrated tasks. On the HP3000, these "service stations" are thought of as System Resources, and three are of primary importance when evaluating system performance. The three main resources that are typically used are CPU, Disc I/O, and Main memory availability.

The concept of resource utilization is nothing more than analyzing each of the resource to determine how heavily it is being used. Every resource has a defined capacity which specifies how heavy a workload it is capable of supporting. For the CPU it is calibrated in the number of instructions per second, for Disc the measure is number of I/O's which can occur per second, and memory capacity is measured in how many bytes of data may be resident. A utilization analysis reveals how close to capacity the resource is operating. The closer to capacity a resource is being utilized, the more significant an impact on performance that resource has. When a resource is being utilized at or near its' capacity, it usually, becomes a bottleneck to overall system throughput.

To understand the impact resource utilization has on overall system performance, a brief discussion of Queue operation is in order. A queue is nothing more than an ordered line. Queues are the way the computer keeps track of those processes waiting for a resource. An example on the HP3000 which everyone is familiar with is the SHOWQ command. The right hand column of the SHOWQ command provides a list of all processes currently waiting for the CPU, essentially a dump of the CPU queue.

Figure 4 provides an illustration of the operation of a queue.

Figure 4



In this diagram, users arrive at the resource (server) to request service. The amount of time spent at the resource (service time) is calculated by adding the time spent waiting (queue time) to the time necessary to perform the service (access time). Not only does the work the process is requesting to be done have an impact on response time, but the amount of time spent waiting for the resource can have a significant impact.

Queue wait time is directly related to server utilization. As server utilization increases, the number of processes waiting in line also begins to increase. When the utilization rate rises above 75-80% of capacity the queue length increases significantly.

Figure 5

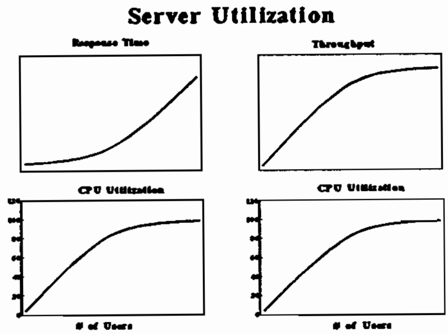


Figure 5 shows a collection of 4 graphs which illustrate the effect CPU utilization has on response time and throughput. These graphs all show the effect the addition of users has on response time and throughput.

Analysis of the response time graph shows that there is a point where the slope of the curve changes significantly. Before reaching that point, the slope of the curve was very small, indicating a gradual increase in response time. Once the point was crossed each additional user caused response time to increase significantly. This point, also known as "the knee of the curve" typically occurs when the resource is being utilized greater than 75-80% of capacity. This increase in response time illustrates the impact of Queue wait time. Once the resource has reached the 80% utilization level, the processes are spending increasing amounts of time simply waiting for the resource.

This same concept holds true when considering throughput. Even though additional users continue to be added, the throughput curve becomes very flat once the CPU is greater than 80% utilized. This indicates that even though more workload is being added, the amount of work being accomplished is diminishing due to the unavailability of the resource.

In the next section more time will be spent on how to use these concepts to manage performance. In addition to this brief overview, there are many ways to increase performance knowledge. First and foremost, experience is the greatest(although not always the most enjoyable) teacher. Besides good old OJT, there is a vast array of reference material dealing with performance concepts. The many HP periodicals(Interact, Chronicle, HP Professional, SuperGroup, Etc.) frequently contain excellent articles concerning performance, and their utilization is highly recommended. Training and performance consulting are also available from a number of sources.

Developing a Plan

To begin developing a site specific performance management program, the first step is to make sure the current environment is understood. There are three areas which we should gather information on:

1. Current system performance metrics
2. Existing performance maintenance procedures
3. Corporate business objectives - longterm plan

In order to document the current performance metrics, data needs to be collected in the areas of response time, throughput, and resource utilization. To gather this information there are three methodologies which can be used: contract with a consulting service, utilize a performance management tool, or use basic MPE capabilities coupled with elbow grease.

How to monitor using MPE will be discussed in more detail in the monitoring section, but briefly some of the activities that would take place include:

- Measuring response times with a stopwatch
- Calculating job throughput by job number
- Using application counters to measure order rates
- Use Resource Accounting(Report command) to calculate average CPU utilization
- Use log file analysis(file opens/closes) to approximate I/O rates
- Use log file analysis to calculate throughput levels
- Use disc caching to calculate physical I/O rates

The easiest method of collecting performance information is using a performance management tool. A tool such as LaserRx provides the ability to answer all the performance background information desired. The ability to collect performance measurements continuously and then review at a later period in time is a key benefit of tools.

Whether the data is accumulated via a MPE & elbows, a tool, or a consulting service the goal is to analyze the current situation to ensure that a performance bottleneck is not currently being experienced. This allows the program to be started on a clean slate without existing problems.

Detailing existing performance maintenance procedures is important to insure that they are included in the newly created plan. It is also good to review plans to insure that they are still applicable in the evolving environment in which we work.

The third task in the startup phase is to define the corporate(or local governing body) objectives and determine how they impact the MIS objectives. As companies rely more and more on their information systems, it becomes increasingly important to take corporate growth into account during the system planning cycle. Laying out the corporate growth goals will become very important as the performance monitoring, analysis, and planning scheduled is developed.

By defining the current environment and laying out the corporate objectives we know have two thirds of the plan defined. We know where we are, and we know where we want to be, the rest is simply defining how we are going to get there.

Creating a Control Structure:

The concept of Service Level Agreements discussed earlier is the key to the success of a performance management program. SLA's provide the goal to strive for, as well as the measurement of success.

The development of a Service Level Agreement is a fairly simple process. Essentially all that is required is to compile the system expectations from both the user community and the management team. Once these expectations have been compiled, the next step is to break them down and convert them into computer measurement terminology. The dialogue below provides examples of this process:

MIS: ***"What is the most important service the system provides for you, and how do you expect that service to be delivered?"***

User 1: ***"I expect the Inventory availability report to be on my desk by 8:00 A.M. every morning."***

User 2: ***"I expect never to have to wait more than 5 seconds to pull up an order. This is extremely important when I have a customer on the phone."***

User 3: ***"I expect to be able to logon to the system every morning at 7:00 A.M.."***

Manager: ***"The system needs to be available until 12:00 A.M. every evening for my second shift foreman to validate attendance."***

All these are very valid needs, and have been expressed in terms that are meaningful to the user. M.I.S. now has the task of converting these needs into a metric that is measurable by the system. Examples of how these might be re-stated include:

User 1: Nightly batch finished by 5:00 A.M. every night. This can be verified by the nightly job log. The 5:00 A.M. cutoff was chosen to allow for printing and distribution of reports.

As an aside this type of Service Level can lead to trade offs in the planning process. A better response to the request might be to install work group printers, to more efficiently route the report.

User 2: The maximum acceptable response time for order entry is 5 seconds, with an average expectation of 3 seconds or less. This metric can be validated using a tools such as HP Glance or LaserRx.

User 3: System availability at 7:00 A.M.. This metric can be verified using the job logs.

Manager: System availability until 12:00 A.M.. This metric can also be verified using the job logs.

Once the quantification has been made, an agreement can be drawn up which would look something like this:

System Availability:	7:00 A.M. - 12:00 A.M. Validated by analyzing daily job logs as well as process statistics of LaserRx.
Nightly Batch:	Complete by 5:00 A.M. Validated by analyzing daily job logs as well as process statistics of LaserRx.
Maximum Order Entry Response Time:	5 seconds or less. Verified using LaserRx.
Average Order Entry Response Time:	3 seconds or less. Verified using LaserRx.

Before finalizing the SLA, there is some additional information that probably should be gathered. The implications of not achieving the desired service levels can be very useful information when weighing planning alternatives. Implications such as:

- Lost orders due to unhappy customers (response time)
- Bad inventory decisions (old inventory report)
- Overtime necessary to make up work (system availability)

are very good input when building justification for the planning alternative chosen. They also provide a good forewarning of the consequences of not achieving the specified SLA.

A last piece of information to gather is how fast the users expect to grow. This information helps facilitate the planning process, as well as providing some checks and balances in the service level process.

Monitoring system activity

Once the service levels have been decided upon, and M.I.S. has agreed that they are all currently deliverable, a measurement schedule must be devised. The measurement schedule documents at what intervals the system will be measured to determine service level achievement. Monitoring schedules are typically determined based on the growth rate of the system being analyzed. System growth rates can be roughly classified into three basic categories:

Stable:	Very slow steady growth with little or no abrupt changes.
Active:	Steady consistent growth.
Dynamic:	High unpredictable growth.

The appropriate monitoring schedule will vary from site to site, but some rough guidelines based on the type of growth environment are provided below:

Stable: every 6-12 months
Active: every 2-4 months
Dynamic: every 2-4 weeks

Along with determining the monitoring schedule, a monitoring methodology must also be chosen. As stated earlier performance tools provide the best mechanism for collecting on going evaluation information. A tool such as LaserRx provides many features designed to facilitate the performance management process. Among the key features to look for in a tool:

- Continuous data collection
- Global Resource utilization information
- Transaction throughput information
- Response time information
- Bottleneck detection
- Application specific data collection

There are a number of tools available which all provide the features necessary to manage performance. Choosing a tool to help monitor the performance levels being achieved is the most time and cost effective way to perform the task.

However, if one does choose to not use a tool, the capabilities of MPE can be used to provide much of the resource utilization information. One of the MPE methods mentioned earlier in the paper was Resource Accounting. Resource Accounting is the information that is supplied whenever the REPORT command is executed. Among the data provided via the REPORT command, is the total CPU utilization by GROUP and ACCOUNT. One of the ways to use this data, is to collect REPORT command information over time, and use it to calculate an average CPU utilization. The steps involved in performing this calculation are as follows:

1. Establish a report command starting point. Either perform a RESETACCT,CPU or perform a report command to a file.
2. After a known time period execute another REPORT command.
3. Total the values in the CPU column for the two command executions.
4. The difference in the two totals is the number of user CPU seconds measured during the time interval.

5. Dividing this number by the number of seconds in the interval provides an average CPU utilization for the interval.

This process can be enhanced by performing the report command to disc files and then downloading the disc files to existing spreadsheets. This will allow the automation of the calculations, as well as enhancements to provide utilization levels at the ACCOUNT or group level.

Using resource accounting in this manner does have a number of pitfalls which should be noted. The most significant shortfall is that CPU seconds are only updated at session/job logoff time, so depending on when the report command is executed the results could be significantly different. The labor required to perform this method of CPU analysis is also very detrimental. Despite these shortfalls, Resource Accounting does provide a decent method of approximating average CPU utilization.

Another resource that needs monitored is the disc I/O subsystem. On MPE V the disc caching statistics can be utilized to provide average I/O rates for both the system and for particular applications. Figures 6 and 7 provide examples of how these calculations work.

Figure 6

Sample Showcache Output – Calculate System IO's/Sec

Calculating how many **I/O's per second** your system is actually doing is most easily done using :SHOWCACHE. Enter the command twice, first when you expect the system to be heavily used, and again an hour later.

:COMMENT - THE FOLLOWING WAS DONE AT 1 PM:
:showcache

DISC LDEV	CACHE REQUESTS	READ HIT %	WRITE HIT %	READ %	PROCESS STOPS	K-BYTES	% MEMORY	CACHE DOMAINS
1	1351440	87	79	73	230981	615	7	147
2	1111146	86	48	64	133972	1117	13	514
3	1143689	86	52	64	156994	2430	30	901
TOTAL	3606275	86	59	67	521947	4162	51	1562

58% of user I/O'S eliminated.
Data overhead is 413K bytes.
Sequential fetch quantum is 96 sectors.
Random fetch quantum is 16 sectors.
Block on Write = NO.

:COMMENT - THE FOLLOWING WAS DONE AT 2 PM:
:showcache

DISC LDEV	CACHE REQUESTS	READ HIT %	WRITE HIT %	READ %	PROCESS STOPS	K-BYTES	% MEMORY	CACHE DOMAINS
1	1423440	87	79	73	230981	615	7	147
2	1147146	86	48	64	133972	1117	13	514
3	1161689	86	52	64	156994	2430	30	901
TOTAL	3732275	86	59	67	521947	4162	51	1562

58% of user I/O'S eliminated.
Data overhead is 413K bytes.
Sequential fetch quantum is 96 sectors.
Random fetch quantum is 16 sectors.
Block on Write = NO.

Calculate the number of physical IOs that had been done at 1PM and again at 2PM. Then take the difference to figure out how many had been done during the hour. Divide by 3600 seconds per hour, to determine the number of IOs done on the average per second.

NUMBER-OF-PHYSICAL-IOs = NUMBER-OF-CACHE-REQUESTS * (100% - PERCENT-ELIMINATED)

2PM: 1,567,555 = 3,732,275 * 42%
1PM: 1,514,635 = 3,606,275 * 42%
52,920 I/Os done during the hour = 14.7 I/Os per second

Figure 7

How Many Disc I/O's Does It Take To Process One Transaction?

- o Logon to the System at a time when you have it all to yourself (no other users - no batch jobs!)
- o Run the Program, Display the Transaction Prompt
- o At another terminal, type: SHOWCACHE & note the number of Cache Requests
- o At the 1st terminal, do one transaction
- o At the 2nd terminal, do another: SHOWCACHE
- o Subtract the numbers to find out how may I/O's it takes to do 1 transaction

A final source of performance data is the system log files. Information concerning job/session logon rates, process creation rates, file opening/closing rates, I/O activity and so forth are all contained in the system log files. MPE provides a couple of tools LISTLOG5(MPE V) and LOGTOOL(MPE XL) to analyze these files, and there are also a number of third party and contributed tools which perform the same function. An example of where log file analysis might be used in the performance environment is file close records. By analyzing the file open/close records a good approximation can be made towards determining the heaviest accessed files. This information can then be used in creating a file placement strategy.

It has become fairly obvious that there is a significant amount of work involved in acquiring these numbers strictly from MPE. The third monitoring alternative after using a tool, and using MPE, is to utilize consulting services to perform both the monitoring and the analysis phase.

Now that the service level agreements have been established, and a monitor schedule and method has been decided upon, it is time to discuss how to analyze the measurements taken. The next section goes through the most common analysis steps, as well as some of the tuning trade-offs available.

Analyzing System Performance

When analyzing the system to determine which resource is currently the bottleneck to overall system throughput, begin by analyzing the utilization levels of each resource. Provided below are a few guidelines for each resource. These guidelines are meant as general rules of thumb, and due to application and processing-mix differences may require adjustments for a particular system.

CPU ANALYSIS

CPU Busy > 75% — This is a good starting point, but you must also take into account the workload profile of the system. If the system processing is primarily batch oriented, then utilization rates greater than 75% will be common. This is particularly true under MPE/XL where even one or two background batch jobs when running with on-line users may cause the CPU utilization to approach 100%. This is primarily true due to the effectiveness of the prefetching algorithms of the storage and memory managers which attempt to eliminate process pauses for disc I/O. By eliminating the pause for disc, the processing, which under MPE/V may have been I/O intensive, now becomes very CPU intensive.

Long Dispatch Queue — The dispatch queue is a dynamic list of processes which are either waiting for the CPU, or are in a "Short Wait" state (for example, waiting for a memory manager I/O to complete). Consistently having 8 or more processes in this queue for a lower-end system or having 10 or more for a higher-end system (950+) could indicate a CPU bottleneck and may require further investigation with more detailed tools. A simple way to determine how long the dispatch queue is at any given moment is to use the SHOWQ command. The right hand column lists all processes currently in line for the CPU.

CPU bound application — An application that is CPU bound has the CPU as the limiting factor to performance. CPU bound applications can be analyzed by monitoring the Average Short Transaction time (AST) or System Average Quantum (SAQ) as well as monitoring the application with GLANCE or

LaserRX. The AST (MPE/V) or SAQ (MPE/XL) give an indication of the average amount of CPU required to complete an on-line transaction. The larger the number, the more CPU intensive the on-line transactions (on average). If a CPU-bound application is causing delays in other on-line processing, artificially lowering the AST/SAQ with the TUNE command will cause the CPU-bound process to drop to the limit of it's executing queue (CQ, DQ, etc) at a faster rate. Improvements to a CPU-bound application require either application modifications or a CPU upgrade.

DISC I/O ANALYSIS

Paused for Disc > 15%

On an MPE/XL system or an MPE/V system with disc caching, if the amount of time spent paused for disc is greater than 15%, then the disc I/O profile should be investigated further. Paused for disc indicates that the CPU is idle because all processes which could be using the CPU are waiting for a disc I/O to complete. This could also be an indication of possible memory bottlenecks.

Disc I/O Rate

If the disc I/O rate rises above 15 I/O's per second for any single drive on the system, that drive may begin to become a bottleneck. Action should be taken to spread the I/O evenly across the available disc drives.

On an MPE/XL system, if the number of I/O's for the system is greater than 30 per second, further analysis should be performed concerning data and code locality.

MEMORY ANALYSIS

Memory Management CPU > 5%

If greater than 5% of the CPU is being spent on memory management activity, memory may be becoming a bottleneck.

Memory Management Disc I/O's > 5

If more than 5 disc I/O's per second are being spent for memory management I/O, memory utilization should be further analyzed.

Plan

After examining the current resource usage levels of the system, there are two possible conditions: either current performance is meeting the objectives of any service level agreements or other performance level criteria, or current performance is not meeting expectations. Depending on this condition, decisions can be made on possible courses of action.

If performance is meeting or exceeding user expectations and agreements, no action may be required. However, if current resource usage levels are increasing or planned business growth will affect system usage, capacity planning or forecasting system usage should be performed. The goal of performance management is "no surprises", and by taking into account the current and possible future business climate, additional system requirements can be anticipated and planned rather than discovered in a "crisis" mode. Through capacity planning or forecasting, you will be able to anticipate future system loading and possibly begin to take any of the action steps outlined below. In addition, the forecasting will allow you to concentrate your efforts on those resources which have the greatest potential for becoming a future bottleneck. Tools from Hewlett-Packard as well as third parties can assist in forecasting future performance based upon historical data. Also, consulting services are available for more in-depth capacity planning functions.

If current performance is not meeting objectives from any service level agreements, we have two problems. First, some action must be taken to correct the discrepancy. This could be any or all of the following:

- Analyze applications and system processing schedules to effect better utilization of resources. This may require application modifications, better management of batch processing (moving non-critical processing to off-hours), or database tuning. Refer to the next section on Tuning Guidelines. Tuning efforts should always begin with the simpliest to implement first - usually they will have the greatest impact as well. At some point the effort to implement a change out-weighs the expected results and additional avenues must be investigated.

- Purchase of additional hardware/software to alleviate resource bottleneck. Although this is usually the last resort for many customers, this may be the most cost effective when you take into account the time and costs of making application modifications to improve performance.

- In agreement with the user community, adjust the expectations set forth in the service level agreements to reflect the current throughput of the system - lowering expectations of the users. This may be the only option for those sites with no development staff or third party software and limited budgets.

Secondly, if performance is not meeting objectives, and this condition was not anticipated, better forecasting and planning measures may need to be put into place for the future, since the goal is to never be in this situation. This is the same type of planning that must take place if the business conditions are changing and may require special forcasting tools or performance consulting.

Tuning Guidelines

If the decision has been made to tune resource usage for better utilization, either in anticipation of future requirements or due to current performance problems, the following guidelines will be of assistance. It must be noted that not all of the guidelines apply to all systems and that cost/benefit analysis may be required to determine if the effort of implementing a change is worth the potential benefits in resource savings.

<u>CPU Management Guidelines</u>

Terminal Baud Rate. Set terminal baud rates at the highest possible speeds, typically 19,200 for direct connects. Although this does not reduce CPU loading, it does increase the perceived speed of the system by the end users. Also, if running under MPE/V, move busy terminals and printers to ATP ports since there is less system overhead associated with ATP I/O as compared to ADCC I/O.

Limit excessive logons/logoffs. This is one of the costliest activities on the system, particularly under MPE/V. Not only are system tables initialized and system space allocated, various processes are initiated, files opened and UDCs initialized. Part of the solution may require a menu system be put into place, and will likely require careful planning with regard to security issues and file access rights. Under MPE/V there are a number of utilities which allow the user to change logon user & account without incurring the overhead of actually re-logging on (ie. TMPCONVP.CONVALL.SYS).

UDC management. Ideally, maintain at most only 1 UDC file for each level (System, Account, User) to limit the amount of directory searching and file opens which must be performed. Also, block UDC files and COMMAND.PUB.SYS large (MPE/V) to reduce the amount of physical disc I/O's.

Program Allocation. Allocate frequently used programs on MPE/V (compatibility mode programs on MPE/XL). This will reduce the LOADER initialization and setup time when running these programs. However, remember that allocating a program only benefits the first user of the program. As other users execute the same program another user is already running, there is no additional benefit to having the program allocated.

Use new MPE/XL commands. Where appropriate, use the new MPE/XL CI commands, such as COPY or PRINT instead of FCOPY. This will reduce the amount of CPU/disc required for the operations.

File Extents. Under MPE/XL "condense" files with many extents into files with a smaller number of extents. This could be accomplished with BUILD/COPY or Native Mode STORE/RESTORE. When a file has a large number of extents (possibly thousands), B-Tree structures must be initialized when the file is first opened. This is a time consuming operation and also having a large number of smaller extents tends to reduce the effectiveness of the storage manager's file prefetching algorithms when reading sequentially.

TurboIMAGE Maintenance. These are meant as guidelines only, and vary from database to database:

- Use DBLOADNG or similar utility to understand database storage efficiency.
- Set BUFFSPECS=50(1/120) for all databases. Set to the maximum for the heavily used databases.
- Monitor master dataset capacity (50-80% full).
- Repack detail datasets regularly along primary path, especially for dynamic datasets (insure accuracy of primary path designation).
- Disable ILR on MPE/XL
- Delete unused keys/paths in datasets
- Remove outdated dataset entries
- Enable AUTODEFER for batch processing if standard recovery in the event of a failure is to RESTORE and re-run.
- Reorganize detail datasets after a large number of deletions.
- Use the '*;' item list after establishing initial item list.
- Place sort items at the end of the data entry (record).

Batch Jobs. Limit batch activity which competes for the same resources as interactive sessions. Depending on the resource, batch processing priority will be boosted to a higher priority if the batch process is holding a resource that a higher priority process is waiting for. This is true of some of the TurboIMAGE resources.

Queue TUNEing. Use the TUNE command to reallocate CPU usage and to control priority decaying.

CPU Application Guidelines

Program Optimization. When compiling in Native Mode under MPE/XL, use the compiler optimization features of the language after the program has been developed and debugged.

VPLUS. When using VPLUS, compile forms into a Fast Forms File within FORMSPEC.

Limit FOPEN/FCLOSE. In both the MPE/V and MPE/XL environments, opening and closing a file are very expensive in terms of resource usage. In designing an application, limit the frequency of file opens and closes to reduce both CPU and I/O activity.

Limit Process Creation. The creation of a new process, like opening a file, is very expensive and time-consuming. If possible use the CREATEPROCESS intrinsic to initiate the child process, and when control must be returned to the parent, SUSPEND the child rather than terminating if the child will ever need to be executed again. This will save on initialization overhead, and increase users perceived response time on subsequent executions of the child program.

Subroutines vs CREATEPROCESS. Under MPE/XL Native Mode, the MPE/V code and data segment limitations have been eliminated. Whereas under MPE/V multiple programs may have been developed to provide the necessary functionality in an application, under MPE/XL these multiple programs could be combined into a single large program. This will eliminate the overhead of process creations when switching application functions. This strategy could also be implemented in a menu system where each application function could reside in an XL module. This has the added benefit of allowing data files, VPLUS forms files and databases to be opened once by the driver program and accessed by the various subroutines.

OCT CM Programs. Under MPE/XL, Compatibility Mode programs should be object code translated. This is a very simple action, and does not require the source for the programs. Depending on how much time is actually spent in the user CM code, significant savings in CPU usage are possible. One side-effect to be aware of is that since Native Mode instructions are actually appended to the tail of the Compatibility Mode code, the file size of the new program file will increase anywhere from 2 to 10 times the original file size.

Migrate to NM? This is an easy statement to make, but difficult to determine if it really makes sense for an application. As has been documented in a number of case studies, migrating an application to Native Mode could actually cause a program to use more CPU than if it was left in CM. This could be true if the program is making extensive use of system services which are still functioning in CM. This includes such functions as MSG, CIR and RIO files, extra data segments, and up until release 2.1, KSAM and the spooling system. By migrating to Native Mode, these applications still made use of the CM operating system services, but now also incurred the overhead of a context mode switch (NM to CM). However, eventhough an application may use these services, depending on the amount of time spent in the user code and the number of procedure calls to these services, an application may still experience significant improvement. The bottom line is, migrate those applications and programs to NM first that make the most sense. Those most frequently accessed and spend the most time in user code. System LOG file analysis could assist in determining the most frequently executed programs.

Mapped File Access. Mapped file access is a new programming tool with MPE/XL. A number of articles have been written in both the Interex proceedings and Interact magazine describing the potential benefits and pitfalls of using mapped file access and therefore it will not be discussed in depth here. However, if CM programs are using extra data segments (XDS) and are being considered for migration to NM, use mapped file access instead of the XDS routines. There exists a set of native mode subroutines to transparently access a mapped file using the same GETDSEG/FREEDSEG/DMOVIN/DMOVOUT calls as on MPE/V.

Use optimal data types. When migrating to Native Mode or developing new NM applications, use 32 bit alignment and binary data types for structures and work variables where ever possible.

Use Transaction Manager instead of FCONTROL. Under MPE/V, FCONTROL (6) was sometimes used to force FWRITEs to disc to guarantee integrity. Under MPE/XL, use the Transaction Manager to guarantee the physical integrity as well as reduce the number of physical I/O's by performing gathered writes.

Disc Management Guidelines

Disc Caching. Disc caching is a way of trading CPU and memory resources to increase the disc I/O throughput. On MPE/V, the disc caching parameters can be adjusted to improve disc throughput. As a rule of thumb, the amount of CPU used to manage disc caching should be smaller than the number of I/O's being eliminated. If this is not true, it may be beneficial to disable disc caching for 1 or more of the drives, or consider investing in controller caching. If the system is short of CPU or memory, disc caching could adversely affect system throughput.

File Purges. Limit mass file purges (PURGEGROUP or PURGEACCT) to non-peak times since these operations generate a potential large amount of disc activity, since the areas on disk are physically over-written for security reasons.

UDC Management. Block UDC files and COMMAND.PUB.SYS to a large blocking factor (100) to reduce the amount of physical I/O, especially during the logon process.

HP-IB/GIC loading. In general, do not place more than 3 disc drives on a channel, depending on system I/O rates. On MPE/XL use HP-FL (fiber link) connections to allow more drives per channel and to allow for future enhancements.

Reloads. Periodically RELOAD the system to recover any lost disc space (MPE/V) and re-organize disc free space into contiguous regions. Eventhough under MPE/XL files are not limited to 32 extents, there is a possible performance penalty during both FOPEN and file access times. Under MPE/V, the VINIT/Condense utility may also be used to re-pack the disc drives on a drive by drive basis. When RELOADing, care should also be taken in placing the heavily access files across multiple drives to minimize the contention on a drive or set of drives. RELOADing under MPE/V will also place the system SL near the system directory, reducing the amount of time necessary to access these two high activity areas.

Balance I/O Load & File placement. With analysis of performance tools and/or system LOG files, it can be determined which files are the most heavily access on a system. With this knowledge, the system manager should attempt to physically locate these files on separate drives to reduce contention.

Disc Free Space. Manage file space to try and maintain 15-20% free space on the system. Periodically remove outdated files, unnecessary LOG files, editor 'K' files, etc. Use the date options on the STORE command to remove obsolete/outdated files, but be careful not to remove only a few datasets of a database.

TurboIMAGE. Review the general guidelines from the CPU management section. These same recommendations assist in reducing the amount of disc I/O as well as reducing CPU utilization.

Memory Guidelines

Program Segmentation. Under MPE/V a program file is composed of one or more code segments, each segment 16k or smaller. The recommendation is to try to keep the code segments in the 8k and under size to allow better memory management. With MPE/V, the memory manager's unit of work is the segment, therefore, if the majority of segments are of about the same size, locating a section of memory becomes a simpler task, since it would generally require only finding one older segment to overlay. However, if there were a number of smaller and larger segment sizes, the task of loading a large segment may require a large amount of work to consolidate and reorganize various regions of memory to make room for the large segment.

Native Mode Locality Options. The idea behind segmenting programs (CM) and controlling program locality (NM) is to minimize the work of the memory manager in controlling which pieces of code need to be in memory in order for a program to execute efficiently. When a programmer specifies segmentation or locality sets, a determination is being made that for the majority of time the program will be executing in that portion of the code. The use of locality sets can reduce the memory manager requirements and not only improve an individual program's performance, but help reduce the overall system I/O and memory load.

TurboIMAGE. Review the general guidelines from the CPU management section. These same recommendations assist in reducing the amount of memory management, disc I/O as well as reducing CPU utilization.

Where To Begin...

When setting out to begin a performance management program, the first step is to establish some performance goals for the system. These can be as simple as informal discussions with a few users or developing and documenting more formal agreements with the users. In any case, without some sort of target to shoot for, we would not know where to begin aiming. This also helps to set expectations in the minds of the users of which the MIS group is aware.

With some goals in place, the next activity is to establish periodic performance monitoring activities. The frequency of monitoring depends on the activity levels and growth plans for the system. The more dynamic the environment, the more frequently monitoring will typically take place since there will be less reaction time should a potential problem be identified. Also, the activities should concentrate in two areas. First, collect data for later bottleneck analysis and second, collect data which can be related back to the performance goals of the system.

Next analysis of the collected data must be performed. Using the guidelines presented here as well as others gained through your experience, the current usage levels of the various components of the system should be determined to identify any possible limiting factors to system throughput and/or response time.

After analyzing the information, a plan of action should be developed. Depending upon the state of the system and how well the defined goals are being met, actions may need to be taken to correct an identified problem or possibly avert a future problem. As part of comparing current performance levels to the goals, a report documenting delivered service levels should be delivered to management and the service level agreement users.

Managing performance is a continual and evolving process which must be periodically reviewed to insure its effectiveness. Changing business conditions may require modifications to either the performance goals of the system or the monitoring and planning activities. With this process in place prediction and avoidance of performance problems become a more realistic objective.