Hewlett-Packard
Business Users Conference

INTEREX

ORLANDO

PROCEEDINGS

August 7-12
1988

VOLUME 3

INTEREX

the International Association of
Hewlett-Packard Computer Users

Proceedings

of the

1988 Conference of

HP Business Computer Users

at

Orlando, Florida

August 7-12, 1988

VOLUME 3

William M. Lyons, Editor

Business Communications Under New Wave

Alison McCallum-Varey
Hewlett Packard
Office Productivity Division
England

Introduction

What this paper talks about

The aim of this paper is to look at the communication facilities of New Wave - in particular New Wave Mail. By way of introduction, the paper examines how business communications and business systems integration have developed into New Wave. The rest of the paper then focuses on New Wave Mail.

Contents

1. Development of Business Communications

Computers have long been recognized as an important business communication medium. The need for information distribution and messaging systems has grown as users gained access to firstly to terminals linked to host computers and more recently to PCs. Whereas previously memos, documents and reports were distributed in paper form around an organization, now the computer can take over the production and distribution of that information.

Hewlett Packard has been working in this area for a number of years and has produced a series of products aimed at satisfying customers' requirements for information distribution and messaging facilities.

HP DeskManager

HP DeskManager is a well-established information distribution and messaging system for the HP 3000. It has been substantially developed since its introduction in 1981 and now provides a comprehensive range of facilities. Documents, memos, graphics and spreadsheets can be sent across a network. Users can send, reply to, read, print and file their messages. HP DeskManager also gives them access to bolt on products such as HP Schedule for resource scheduling and HP File/Library for community filing. HP DeskManager provides all these facilities on an HP 3000 and is a terminal-based solution.

AdvanceMail

As PCs have become more prevalent, users have moved towards using PC-based applications for creating word processing documents, graphics and spreadsheets. What has not changed is their need to distribute that information within their workgroups and across the company.

To give PC users the same information distribution and messaging facilities as the HP 3000 users, Hewlett-Packard introduced AdvanceMail. AdvanceMail allows PC users to send documents, spreadsheets, memos and graphics across the HP DeskManager network without leaving the PC environment. They also have access to personal filing, printing and automatic conversions. AdvanceMail is a major element in an integrated PC solution to users' information distribution and messaging needs.

New Wave Mail

As the move towards PCs continues, so the need for a even more integrated solution has become apparent. This has now come in the form of New Wave and with it New Wave Mail which provides the capabilities of HP DeskManager and AdvanceMail but in a New Wave environment. Before looking in detail at New Wave Mail, let's examine how the New Wave environment developed.

2. Development of the New Wave Environment

New Wave is the culmination of a long process of computer integration. The concept of integrating systems has been around since minicomputers were introduced. At that time the focus was on integrating the minicomputer with the mainframe. The end user was isolated from computing resources. When PCs were introduced in the early eighties, minicomputers were by that time well integrated and PCs were an island without any form of connectivity. Vendors like Hewlett-Packard recognized the value of PC integration and worked to provide for the needs of the PC user. In the case of Hewlett-Packard this has resulted in our being rated #1 in PC integration by all the leading consultants.

Not content with that, Hewlett-Packard have been investigating how to break down the final barriers between user and computer. It is the aim of the New Wave environment to break down those barriers.

Focus on Tasks not Tools

In investigating what constituted the barrier and how to break it down, one of the facts revealed was that using computers to complete tasks often means focusing more on the tools to do the task rather than the task itself. This means that users productivity is not as high as it could be. To make users more productive they need a computer solution which focuses, as they would do, on the tasks not the tools.

The idea behind New Wave is that users should no longer have to think in terms of electronic mail, graphics, spreadsheets, word processing but rather be free to concentrate on their key tasks - information management, analysis and communication. In the rest of this paper, the emphasis will be on the third task - communication.

In order to make the task-orientated environment a reality, some key elements need to be included. These are:

Improved integration between applications

This integration should be:

Seamless - when producing documents which have elements of text, analysis and graphics, the system should allow users to switch back and forth across a number of applications without complicated commands.

Transparent - Data should only have to be entered once but be presented in a number of different ways without rekeying.

Hot Connects - Different views of the same data should be linked. So when one version of the data changes, so other linked versions changed automatically.

Interrupt-driven - Most people are usually trying to get several tasks done at once. They should be able to move quickly from one task to another without complex commands.

Consistent, predictable user interface - Users should have a single interface which means they only have to learn one set of commands which will apply across all the applications they use in that environment.

Transparent access to network resources - Access to the network should be transparent. The workgroup needs to be connected to the rest of the organization with a minimum of technical knowledge on their part.

New Wave Mail

New Wave Mail is the internal name for the communications component of New Wave. It allows users in the New Wave environment to communicate with other New Wave users as well as with HP DeskManager and AdvanceMail users without leaving the New Wave environment. In addition it gives New Wave users the capability of communicating with external systems such as Telex, IBM's PROFS and DISOSS and other public and private mail systems via the X.400 standard.

In designing for the New Wave environment as discussed earlier, the emphasis is no longer on the tools but the tasks. In the case of New Wave Mail the task to be accomplished is communication. The concepts of New Wave Mail therefore are to enable the user to carry out that task in the most natural and familiar way possible.

The New Wave desktop provides the framework for every task. New Wave mail exists within that framework as the mechanism for carrying out the task of communication. The user is no longer constrained by the idea of different applications for different tasks. Nor are they restricted by the feature set offered by that application. Everything needed to accomplish a task is immediately accessible from the Desktop.

In the instance of New Wave Mail, users wanting to send an object or series of objects can simply pick up the icon representing the object and drop it onto the Mailroom icon, they can then transfer the messages whenever they choose. In this way, the user is no longer obliged to learn a different set of instructions for each application but has an consistent, easy to learn, easy to use environment where he/she can be productive from the outset.

New Wave Mail Components

New Wave Mail is represented on the Desktop by a series of icons which the user can use when they need to communicate. The two primary icons are the Mailroom and the Intray.

Mailroom

The Mailroom icon is for communication from New Wave to other users on the network. It provides the capability to send any item or group of items from the Desktop out through HP DeskManager to either other New Wave users, HP DeskManager users, AdvanceMail users or even external systems through HP DeskManager's foreign service gateway mechanism.

The Mailroom uses the addressing system familiar to all users of HP DeskManager and AdvanceMail - the user's full name plus a location and sublocation combination. As with HP DeskManager and AdvanceMail, New Wave Mail allows users to send a distribution list of up to 200 users. It also gives the capability to use foreign addressing for delivery beyond the HP DeskManager gateway.

Intray

The Intray icon represents the area to which messages are delivered. The New Wave user receives messages via HP DeskManager and can store them anywhere on the Desktop. Items received are represented as addressed envelopes on the Desktop.

Other Icons

New Wave Mail uses an envelope icon to represent items which are sent and received by the New Wave user.

Outgoing Envelope - The user can group items in an envelope before mailing them. This includes a distribution list and up to 200 items.

Incoming Envelope - Items received in the In Tray are displayed as envelopes when moved to the Desktop.

Returned Envelope - If a message could not be sent for whatever reason it is displayed as an envelope with a cross through it.

Other features

Business communications have tried up to now to provide all the facilities a user might need within the package itself, so that the package becomes the user's working environment. This is an important stage in the development of the integrated environment. The idea of having all applications under one "umbrella".

New Wave replacing that concept and it becomes the working environment. Communications now fit into that environment rather being the environment. New Wave gives access to features such as word processing and filing via the Desktop where they are shared by all New Wave applications.

The features that New Wave Mail provides are therefore those which are directly related to communications needs. These include

Distribution List processing
Since New Wave Mail uses HP DeskManager's transport mechanism, it ensures that distribution lists are correct.

It provides different categories of sender - TO: CC: BCC: and FROM:.

Distribution lists can be prepared and stored in the file drawer and then added to messages as required.

Filters

When users receive large numbers of messages, filters become an invaluable way of saving time by allowing the user to do some of the processing automatically. Users can specify which messages they want to receive and where they want them to be put according to the sender and subject. This means they don't have to deal with "junk" mail and can receive important messages more quickly.

Conversions

When communicating around a large network, it is inevitable that different users will be using different applications, particularly if they are non-New Wave host-based users. The converters reside on the HP 3000 and enable the New Wave user to set up in what format he/she will receive items. HP Draw can automatically be transferred to Drawing Gallery for example. For more details of how new and current wave applications interact see "New Wave and Current Wave - working together".

Message Transfer

Message transfer can take place in both directions - sending and receiving. The transfers can be set for a particular time and sending and receiving can be done at the same or separate times.

Messages are routed around the network through HP DeskManager. In order to take advantage of this facility, the New Wave Workstation has to be connected to an HP 3000 either as part of an Officeshare network using ThinLAN, StarLAN or StarLAN 10 or by means of an HP Serial connection.

Connections can also be made over X.25, modem, HP 2334 multiplexor. Other connections can be established using AdvanceLink.

New Wave and Current Wave - Working together

Hewlett Packard has a commitment to the future and New Wave is the way that we see it developing. However, we are also committed our customers and we understand their need to protect their investment in current solutions.

With these concerns in mind, we have worked to ensure that the current and the new wave can coexist and are by no means mutually exclusive.

Working together - how it's done

Many existing customers will have both HP DeskManager and AdvanceMail users on their systems. Some may be using terminals, others less powerful PCs. They also want to allow some users to move to using New Wave. All the users will have the same need to communicate as they did before New Wave and many of them will be using current wave tools to do so.

To ensure that the needs of all users are served by the new and current wave applications, the current wave applications have been modified to recognize New Wave objects and treat them accordingly and New Wave was designed to accept the current wave objects from the outset.

This has been achieved by developing a series of browsers which enable the conversion of New Wave objects into current wave objects as they are transported around the network. HP DeskManager will contain a Serialized Object Format browser and this will also be utilized by AdvanceMail.

Thus the new and the current can coexist and users of different types can communicate with the minimum disturbance of their established methods.

Conclusion - We've come a long way

The development of New Wave is a major breakthrough in efforts to increase user's productivity. Finally they can concentrate on the job in hand and not be concerned with the task of learning numerous different software packages and applications. It's a long way from where we started and the development doesn't stop here.

In the coming year, more the communications horizon will expand even further. Areas under investigation right now include connections to facsimile, the u of Voice messaging, Unix mail, and EDI Value-added networks. These areas are still being investigated and some not come to fruition, however whichever direction we take, New Wave is our platform for the future development of communications. The world will be at the fingertips of the New Wave user.

Community Filing with PC's

Author: Jon Baker
Version: 2.0
Printed: May 30, 1988

OFFICE PRODUCTIVITY DIVISION

Nine Mile Ride
Wokingham
Berkshire
RG11 3LL

ENGLAND

2002

TABLE OF CONTENTS

ABSTRACT

HP File/Library is HP's first community filing product and is currently available as a "bolt-on" product to HP DeskManager. Since its introduction in 1986, the need to extend the service to PC users has become increasingly important. In addition, customers without HP DeskManager have also expressed an interest in allowing PC users to share information effectively.

The integration of PC's in a community filing environment presents a number of new and challenging problems, these include allowing both terminal and PC users to share a single community file store, integration with PC applications, coping with limitations set by the MS-DOS file store and finding a suitable transport architecture which offers a satisfactory level of performance.

This paper presents solutions we are devising to offer community filing services for PC users. It includes an overview of the product's architecture and a technical discussion of the main design areas including the PC component, the transport mechanism and the interface to File/Library.

In this paper I propose to examine how the PC fits into the world of community filing. It is based on my own experiences from working in this area. For those new to community filing I will present a short summary explaining what it is and highlight the main benefits to end users. Following that I will concentrate specifically on the integration of PC's into such an environment, I will discuss the problems found during design and present solutions we are currently devising.

1.1 Community Filing

Community Filing is quite simple the "*Sharing of large amounts of unstructured information amongst a group of office users*".

What are the commonest types of information that users want to share ? There are three main categories: textual information (the most dominant), spreadsheets and graphics. Note we are **not** infringing into the world of databases, the information we are storing is *unstructured* consisting of many *items* of varying size and format (typically these will be files created by applications).

What operations will users want to perform on this shared information ? Certainly they would want to be able to read it (or *browse*), they may want to use an application to *edit* an item or simply extract selected parts using *cut* and *paste* facilities. They will want to ensure that their information is secure, i.e. they will want to specify who can read or change their items. Users may not require all items to be on-line at the same time, this gives the system the opportunity to *archive* items onto an offline medium such as magnetic tape to save disk space. These archived items could of course be retrieved when needed at a later date. References to non-electronic items such as books or manuals are also becoming increasingly more important. We refer to these as *offline* items. The information shared amongst the group now becomes an index entry which will probably contain a reference to the physical location of the offline item itself.

The scope of the community sharing the information could be any group of people you wish to name. Here are a few examples: A small group of colleagues, typically in the same department possibly located within close proximity of each other. A product team could form a group consisting of a member from each department of a company. Inter-office groups might consist of people within the same company but who are resident in different offices, maybe a group of managers who communicate regularly with each other. A standards committee might have representatives from a number of different companies, the members of this group could potentially be spread world-wide. Remember also that any user may be a member of more than one group. It is common these days to hear these groups being referred to as *Work Groups*.

Before we look at the requirements for a community filing product we need to clarify the basic hardware architecture for the type of system we are discussing.

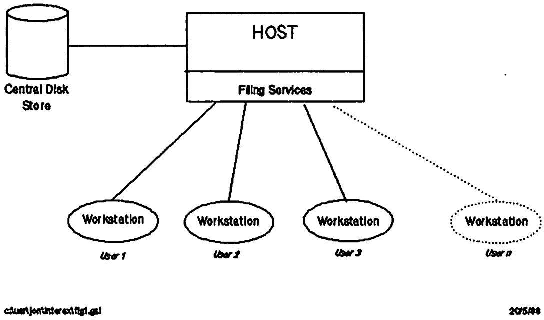


Fig-1.1: Elementary architecture

All the information for a work group is centralized on a host machine, every member of the group needs to be able to connect to this host. The filing services that manipulate this information can support many different groups on the same machine, note that all items belonging to a group are held on this one machine. The workstation could either be a terminal or PC, we will discuss the actual choice of host in chapter 3.

Having looked briefly at what community filing is let us look further at the requirements for a community filing product. To help us I have written three scenarios where community filing could be used.

2.1 User Scenarios

A team of six technical writers are preparing a manual for a new product, they are using a popular PC word processor to create it. Each writer has been assigned a certain area to concentrate on. The manual consists of a number of chapters, they decide that each of these will be held in a separate file. There are many chapters where each writer will want to make a contribution, therefore control is needed over the editing of files to prevent two or more of them attempting to update a chapter at the same time.

A team of lawyers are preparing the defence for a complex law case, they need to perform extensive research on similar cases from the past. Their firm holds electronic copies of transcripts from all the cases that they handle. They wish to perform generic searches on the transcripts to retrieve any information that might be useful. The store of information holding the transcripts is constantly being added to and is rapidly increasing in size.

The personnel department for a large company are in charge of administering employee evaluations which every employee has annually to discuss his/her progress over the year. Each evaluation is held electronically and stored for future reference. Security is of great importance, the company has a hierachical structure to its organization, the personnel department dictates that an individual may only *read* the evaluations of people working directly under him. He/she may not read the evaluations of colleagues who are at the same level in the organization or the evaluations of people working for them. One final security requirement is that the only people to have edit access to an employees evaluation are his/her manager and the head of the Personnel department.

2.2 Host Filing Services

Fast retrieval information from a community file store is the key to success. Todays modern searching techniques are based around the concept of *full-text-searching*, that is the ability to specify a search with respect to the actual content of the information. For example I might specify a search to retrieve all items in the file store that contain in them the words "Hewlett Packard". The specification of searches can be made quite complex by introducing logical operators and other notations. For example I could ask to search for all the items containing *either* the word "Hewlett" or the word "Packard" or both. A full-text-search mechanism works by creating sophisticated indexes that are built as items are added to the store. All the scenarios in 2.1 would need a good searching mechanism particularly the team of lawyers.

Full-text-searching really only makes sense for items containing text, which is probably the majority of information filed. However what mechanism do we use for retrieving non-textual items? One way is to use *fixed-attributes*, these are assigned to every item that is filed into the community store. Typical types of attribute may include Author, Subject, Comments and the most powerful attribute User-defined Keywords. Your searching mechanism now searches on the values of these attributes, for example you could retrieve all items whose author is "Jon Baker". You will of course want to be able to specify searches in terms of both the content of items and the values of attributes combined, for example retrieve all the items with "Hewlett Packard" in the content that were authored by "Jon Baker". Fixed-attributes are the way in which you can implement the indexes to the offline items mentioned earlier. Remember that offline items are simply a reference to a physical item, there is no associated electronic item just an entry with attributes ...

Once you have a retrieval mechanism your next requirement is to be able to describe exactly whom you wish to allow access to your information. This introduces the topic of *Security*. If we define an item as the unit of information within your file store (this would probably be a document, spreadsheet, chart etc.) a comprehensive security system should allow you to configure security at this level. How is security specified ? There are basically two parameters firstly the security attributes for an item i.e. Read/Write/Delete and secondly the scope of the attribute which is specified in terms of a pre-defined group, this group would be configured elsewhere and allocated a name for referencing. The personnel department in the third scenario placed great importance on security with a very complex structure

Editing items was an operation we identified in chapter 1, but how would this work? The actual editing itself is done by the application that originally created the item. The community filing product has to provide a mechanism which allows the application to be invoked and secondly lock the item so that it cannot be accessed by anyone else while changes are being made. It would also be useful if other users could see who was editing a particular item. The technical writers in the first scenario need a good editing mechanism to prevent loss of data through simultaneous updates of the same chapter.

Archiving is an essential feature of any community filing mechanism. With the amount of information that could be filed being virtually unlimited, it is necessary to periodically archive off onto a secondary media (such as magnetic tape) items that are infrequently accessed. But note one important feature, the indexes created for the item (that are used for searching) remain intact, this means that archived items can still be referenced with searches. The user will also need to be able to issue a command to retrieve an item from its archived media back into the store if the item needs to be read or changed. Again all the scenarios could use archiving but the team of lawyers are the best example. They would probably have a very rapid growth in the amount of information being filed into their store.

Terminal users directly access the filing services that are running on the host, they use host based applications to create items. PC's distribute applications onto local machines. This chapter discusses the extra requirements that this places on our community filing product. Firstly let us refine further the type of hardware we intend to use.

3.1 Architecture

From previous experience in building terminal-based filing products we have learnt that 3000 hosts are well suited for locating the core filing services (i.e. storage of the information, searching mechanism, security, archiving, administration and maintenance). This is because of the amount of CPU resource necessary to support community filing together with the ability to connect large amounts of storage. The host itself maybe used for other activities such as mailing, databases etc.

The PC workstations will typically be XT, AT or PS/2 compatibles running MS-DOS or OS/2. They could be linked to the host computer with either an Asynchronus or LAN connection.

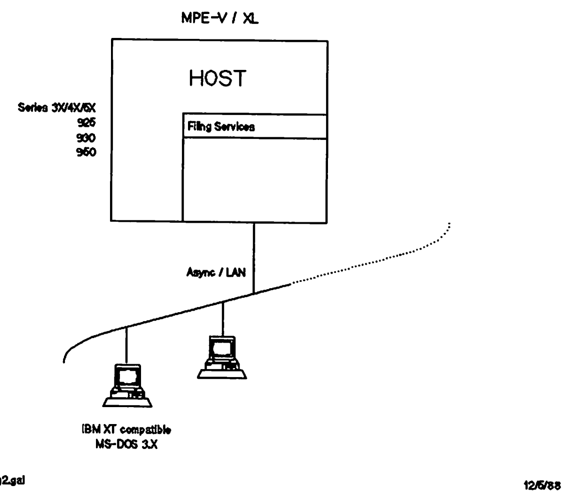


Fig-3.1: Hardware configuration

3.2 Specific PC Needs

PCs off-load host resource onto individual machines which are better suited to dealing with the intensive screen I/O demands of word processors, spread sheets and graphics packages. But they do introduce a number of new problems:

- Applications are now resident on a users local PC rather than being centralized on a host.
- Users now have their own local disk storage to hold files. These cannot easily accessed by other people (unless you use a LAN).
- With all the PC software available there are now many more file formats to deal with.

What additional features do we need to integrate PC users with our host based community filing services?

Interface
We will need an interface on the PC to the host based filing services. This will allow users to access the features we have mentioned already (searching, archiving, security etc.) as well as the item transfer mechanism identified above.

Item Transfer
PC users create items that are held in local storage (i.e. DOS files on floppy/hard disc). With the requirement that the filing services are resident on a host machine we need a transfer mechanism to move items between this and a users PC. This process is likely to be time consuming and should therefore be minimized.

Integration
To work effectively we need a good level of integration between the community filing product and the users PC applications, users maybe retrieving information from the community store to change an item or use parts of it in creating another. This integration typically means using MS-DOS and its file store (most PC applications are invoked from DOS and themselves use DOS files as their unit of store). The integration problem is easier to solve for terminal users since their applications and the filing services can be located on the same machine.

A further requirement is the ability to support *mixed environments* of terminal and PC users. Many companies are in the process of moving terminal users to PC's and therefore may have work groups consisting of both. Conversion facilities will be needed for Terminal and PC users to be able to edit each others items.

3.3 DOS Environment

MS-DOS is the operating system installed on the majority of PC's. Unfortunately it is not well suited to integrating with host based filing services. Firstly there is limited control over DOS files, users can copy them, move them, delete them, or rename them as they wish. There are no unique references to keep track of a file throughout its life (DOS handles are not good enough). To support the checkout mechanism that we described in 2.1 we will need to create a link between the PC and the community file store on the host. When we have no way of tracking a DOS file during its life the link becomes difficult to maintain.

MS-DOS has no support for the coding of files to distinguish their type e.g. Lotus, AdvanceWrite, HP Word/PC etc. MPE for example supports file codes which can be allocated to represent different types of files. Often with DOS the only way to identify a files type is by examining the extension in the file name.

Another feature that would be useful for integrating community filing would be the ability to store some special user defined information in a file somewhere without disturbing its actual content. They could be areas of a file which you can allocate and use as you wish which are ignored by applications when they use the file. This feature would be useful for holding the link information described above.

Many users find difficulties in using MS-DOS, they find some commands difficult to understand and remember (especially from the DOS prompt). Although we are not trying to offer a new shell for PC users we must be careful in the way in which we use MS-DOS in our integration.

In 1986 we released a terminal-based community filing product called **HP File/Library**. **HP File/Library** runs on a 3000 series host under both MPE-V and MPE-XL and the current version is available as an optional bolt-on to HP DeskManager.

HP File/Library consists of a number of flat *catalogs*, created as necessary by users. On entering the Library the user is presented with a list of these catalogs. Each catalog contains the items themselves, they have no hierachical structure (like DOS directories for example). This has the distinct advantage that users do not need to know where to file an item other than the appropriate catalog. The division and retrieval of information within the catalog is achieved through the search mechanism.

HP File/Library can be used to index *any item* held inside or outside of HP DeskManager, including MPE files and *paper* items.

When an item is filed into a catalog it is indexed by 8 fixed attributes which allow you to uniquely identify each item. For items created in or imported into HP DeskManager the user is prompted for three of the fixed *attributes*. These are common to every item indexed in the library : Author, Keywords and Comments. HP File/Library automatically provides a further 5 attributes for each item indexed : Subject, Creator, File Type, Status and Create Date.

Items are *retrieved* from a catalog by searching on one or more of the attributes. Each catalog has an automatically maintained keyword dictionary to help find suitable keywords for searching. Searching on the "Comments" attribute will perform a "full-text-search" on the Comments on every item indexed in the catalog.

HP File/Library offers *security* at three levels (access to the Library, to the catalogs and to the items within them). This allows you to determine what a user sees and how they can manipulate the items that they have access to.

To prevent two or more users trying to simultaneously update the same item there is a *checkout* mechanism. This lets a user copy a document to his work area, edit it and check it back in. While a document is checked out no one else can change it.

Browsers and *conversion* facilities allow users to read and edit documents stored in HP document formats and DCA document formats. A plug in mechanism in HP DeskManager allows users to write their own converters for other document formats which HP File/Library can access.

HP File/Library has an *archiving* facility to allow items to be stored off to magnetic tape. When an item has been archived, references to it are still maintained in the catalog indexes for searching and the item can be retrieved if necessary.

The *next release* of HPFile/Library has several major enhancements including a *full text searching* mechanism on the content of items, in addition to the current searching facilities. There are also plans to provide both a *standalone* version of HP File/Library and a version integrated with HP DeskManager The third major enhancement in the next release of HP File/Library is that it will provide the host filing services for *PC users*. The rest of this chapter discusses how the PC user will be given access to these host services.

4.1 Additional Features For PC Integration

How do we implement the features described in chapter 3 to integrate the PC user with File/ Library?

4.1.1 Interface To Filing Services

We have decided to use the current terminal-based HP file/library interface via AdvanceLink terminal emulation as a simple and effective solution. A full native PC interface would have been considerably more complex, needing a mechanism that could cope with the frequent transfer of information between the PC and the host.

4.1.2 Item Transfer

To initiate the transfer of items between the PC and host we have extended two existing HP File/Library commands. Firstly we have a command to *copy* a PC file into the Library. We have extended the *checkout* command to checkout an item in the Library to a file on the users PC (he supplies the DOS filename). We have a new command invoked from the PC to *checkin* a PC file that was previously checked out from file/library.

4.1.3 Integration

Quick access to the host filing services from applications is essential. The neatest solution is context-switching. There are a number of context-switching products already available on the market (e.g. MS-Windows), typically they allow the user to set up a number of partitions, each partition appears to an applications as a virtual PC with a full complement of RAM. The user can switch from partition to partition by using a pre-defined hot-key. The underlying context-switching mechanism saves the state of a partition when it is switched out to load another. They can use a number of different types of storage to save these states: normal unused RAM, Extended Memory, Expanded Memory and even the hard disc.

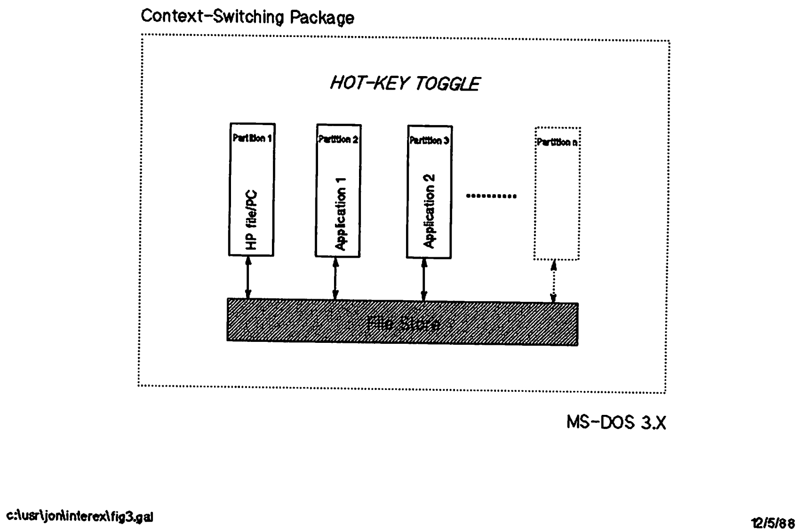


Fig-4.1: Context-switching

We considered the use of a Terminate-Stay-Resident component which could sit under all applications. These however can be unreliable and cause problems. It would also present a large memory overhead to the user.

4.2 Implementation Details

The product has four major components:

Control menu: User is presented with this menu on startup of the product.

Filing services: This component is made up of the AdvanceLink terminal emulator accessing the HPfile/library interface.

Item transfer: There are a number of AdvanceLink command files and some enhancements to the HPfile/library interface to perform the transfer of items between the Library and the PC.

PC controller: PC software that manages the checked-out items while they are on the PC.

There is an interface to this component which allows the user to list all the items he has checked-out on his PC and the DOS filename given to each.

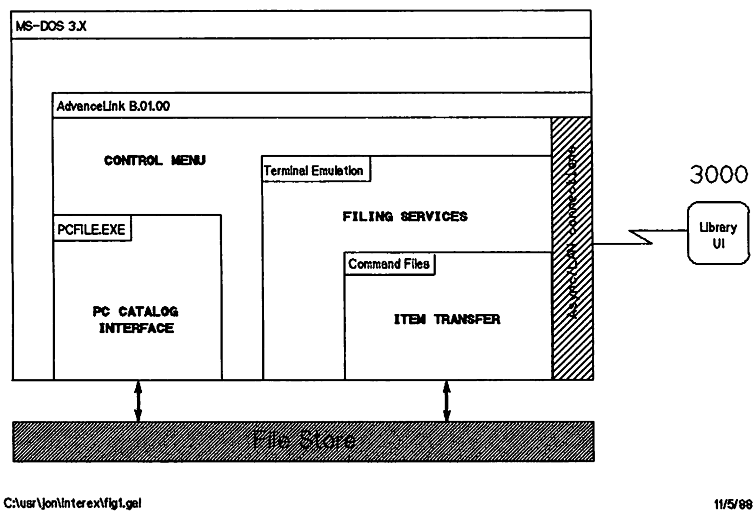


Fig-4.2: Component Integration

4.2.1 Item Transfer

With AdvanceLink being used to provide terminal emulation we decided that we could also make good use of its file transfer and command language capabilities to implement the item transfer mechanism. It can do this across both asynchronus and LAN connections. The performance of the file transfer mechanism is satisfactory, we recognized in our initial investigation that this was always going to be the major performance overhead in the product.

4.2.2 PC controller

This controller maintains a table of entries, one entry for each item checked-out to the PC. Each entry contains information which links the file on the PC with the host store. It contains information to allow us to check the file back into File/Library to where it came from.

This table is also a list of all the items that are currently checked-out on the users PC. We will provide an interface so that the user can view this list and issue checkin directives against selected items.

To minimize the number of file transfers we have included a mechanism inside the controller called *Minimum Data Transfer* (MDT). It works as follows, suppose a user checks-out a large report from the Library, he makes a number of changes and then checks it back in. The copy transferred to his PC when he checked the item out is not deleted after the check-in operation. He then decides he wishes to make further changes to the report. The MDT mechanism discovers that this user has checked the item out before and that a copy still resides on his PC, by using version information that is stored both in each catalog entry in the Library and also a special table created on the PC, it can decide whether the item has been altered by anyone else since this user last checked it in. If no changes have been made no transfer is necessary, the user can use the copy he still has on his PC.

SUMMARY

What is Community Filing ?

- Sharing large amounts of unstructured information amongst a group of users.

What are the key features ?

- centrally accessed store.
- searching mechanism.
- security.
- editing control.
- archiving.

What are the key benefits ?

- fast and effective information retrieval.
- reduce disk storage requirements through archiving and minimizing data duplication.
- restrict access to authorized users only.
- increase the availability of information.

What do we need to integrate PC users ?

- PC <-> host item transfer.
- interface to host filing services.
- application integration.

What solution are we devising ?

- use HPfile/library for host services (searching, security, archiving etc.).
- use AdvanceLink terminal emulation to access these services.
- AdvanceLink command files and new HPfile/library commands for item transfer.
- controller on PC to manage the checked-out items.

NOTICE

The information contained in this paper is subject to change without notice.

TITLE: HP NewWave and Workgroup Productivity

AUTHOR: Bill Crow

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 2003

Using HP DeskManager Intrinsics for Custom Applications

Peter Dunmore
Hewlett Packard
Office Productivity Division
England

Introduction

What this paper talks about:

This paper discusses the integration facilities available in HP DeskManager and introduces HP DeskManager Intrinsics. It describes:

- Integration features available with HP DeskManager and how they can be applied.
  - Script files
  - Application Data passing
- HP DeskManager Intrinsics
  - What are they?
  - Who should use them?
  - When should they be used?
- Three Scenario's where the Intrinsics could provide profit improvements for an organization.

HP DeskManager

HP DeskManager's electronic mail system includes everything the most sophisticated users have come to expect from an integrated business system. Word processing documents, graphics files and meeting appointments can be sent and received across the whole network. To avoid repeated messages and delays, acknowledgments can be set to track the progress of the mail. Users can find out whether their messages have been sent, delivered, read or replied to. Users can also have their mail automatically forwarded to another location. Or, using Auto Answer, they can compose a standard reply which HP DeskManager will automatically send. Users can allow secretaries or co-workers to access their systems as a "designate" - within specified limitations. HP DeskManager includes a high level of system security which reduces the chances of unauthorized access. Access to certain applications can be limited to either system-wide or to specific groups of users.

Every HP DeskManager user has a personal electronic filing cabinet in which documents and messages are filed away in folders. Users can create as many folders as they need (including folders within folders) and nominate other users to share those folders. To ensure privacy users can set passwords on any folder that they have created. All users on the same HP DeskManager node can share Public Distribution Lists and the Notice board. These contain items of interest to everyone, such as commercial distribution lists and weekly newsletters. For more sophisticated filing there is an add-on product called HP File/Library. This is a community filing product which allows documents to be shared by controlled groups of users. Documents are indexed by up to eight attributes: Subject, Creator, Create Date, File Type, Status, Keywords, Author Name and Comments. Keywords and Comments can be added to provide a unique description of an item to help speed the retrieval of that item.

HP DeskManager's time management system ensures that every time a user logs-on to HP DeskManager, the Calendar lists the number of the appointments and ToDo items for that day. More detailed information is available in the Calendar Area, where appointments can be displayed or printed by the day, week or month.

HP NewWave is an advanced application environment which builds upon Microsoft Windows 2.0. It presents the end user with icons on the screen, representing the common tasks people accomplish in an office. With HP DeskManager, we provide NewWave integration by allowing editable document exchange between HP DeskManager, AdvanceMail and NewWave mail users.

There are now over 4500 installations of HP DeskManager worldwide. The product is now localized in nine European languages - German, French, Italian, Spanish, Norwegian, Swedish, Dutch, Danish, and Finnish.

With the B.02.00 release of HP DeskManager last year we have seen greatly improved performance compared to its predecessor, version B.00.00.

- CPU consumption of HP Desk version B.02.00 has been reduced by approx. 22% compared to B.00.00.

- Disc I/O has been brought down to an even greater level - 31%.

HP DESKMANAGER...

- ★ Electronic Mail
- ★ Personal Filing/Time Management
- ★ Application Umbrella
- ★ Task Automation
- ★ Customization
- ★ Forms Processing
- ★ NewWave Integration
- ★ Over 4500 Installations WorldWide

Information Systems Group
INT2

HEWLETT PACKARD

HP DeskManager - A Solution that works the way you work

The last thing you want from an office system is to have to adapt to a new way of working. With HP DeskManager you adapt the system to the way you want to work.

Integration

Variables can be customized at the individual level to allow users to see what they want to see:

- Order items in the Intray and other areas
- Printer and environment file details
- Default editor for text items

Customizable at a system - wide level

- Function key labeling
- Message header
- Welcome banners
- On-screen help information

Benefit:

- Total system is configurable for ease of use
- Saves the user time since trays and displays are ordered, organized and viewed to meet special needs.

Suspend and Resume

Allows users to suspend work in one HP Desk area, move to another, do some work and return to original work location. Just like you would with paper files on your desk.

Benefit:

- Matches the interrupt-driven way we work

Forms Processing

Forms, created with VPLUS/3000, can be distributed, filled in, returned and collated e.g. time sheets.

Benefit:
Saves time by

- Providing electronic mailing and filing for all forms-based tasks
- Not having to re-enter data taken from a data base into a document.

Built in document converters

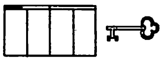
HP DeskManager comes complete with built-in document converters e.g. HP Word to Executive MemoMaker.

Benefit:

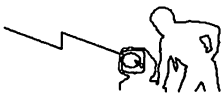
Converters work transparently so users of different word processors can read or print documents without worry. If editing is necessary, the converters perform the task swiftly and easily. Users learn and become productive more quickly.

THE INTEGRATED ENVIRONMENT
HP DESKMANAGER

Simplifies the Interface to the End User



Provides Controlled Access to Applications and MPE



Integrates Communication with the Application

INTEGRATION OF APPLICATIONS AND INFORMATION

* **Add mailing functionality to a HP3000 application.**
* **Simplify and control the user environment.**
* **Make the HPDESK interface available from other applications.**

Foreign Service Connection

Further on in this paper when intrinsics are introduced, the foreign service connection will be discussed again however its worth providing an overview now.

The foreign service connection (FSC) can be used to pass information:

- from an application to users all over the HP DeskManager network
- to an application from users all over the HP DeskManager network
- from one application to another via HP DeskManager

In this context, "application" includes other foreign mailing systems (for example the Telex network or HP Office Connect to IBM Profs or Disoss) as well as HP3000 applications.

FSC uses the widely accepted ARPA file format for the transfer of data. This allows other applications to interpret what it sends.

Application Data Passing

HPDESKMANAGER

APPLICATION INTEGRATION

* Application Area Menu
* Access to Application Controlled by User Group

* Pass Data Between Applications and HPDESK
* Access to Programmatic Document Conversion
* Access to HPLIBRARY

* Forms Processing Capability

* Access to HPDESK HELP Facility from the Application

* Suspend from the Application to HPDESK

HEWLETT PACKARD
Business Computing Systems DINT14

HP DeskManager allows you to integrate your own applications into HP DeskManager. Script Files which will be discussed later and abbreviations allow you to start and direct the running of your applications. However, once you have started up your application, it and HP DeskManager can pass information (files) between each other. This is what is meant by Application Data Passing.

Application Data Passing (sometimes known as the "Applications Umbrella") allows the user of an application program and the application program itself to gain some access to HP DeskManager. The application must always be run from within HP DeskManager, that is, as a SON process.

The user will be in the application most of the time and will occasionally suspend back into HP DeskManager, maybe to read new mail or retrieve a document before returning to the application once again.

The application can access some HP Desk items itself without the user becoming involved. It can also put new items into the HP DeskManager database without user intervention.

The sort of information which can be passed between the application and HP DeskManager includes:

- The identity of the items in the workarea which the applications wants and the name of the MPE file HP DeskManager has put it in.
- The name of the file which is to be passed to the library, and the name of the catalog it is to go in.
- Where HP DeskManager has put the forms batch file the application is to process.
- The help page which is to be displayed.

How Application Data Passing Works

The users starts the application from within HP DeskManager using either:

- the MPE command: RUN
- the EXECUTE command
- an installed abbreviation
- a script file

The application is run as a SON process of the HP DeskManager user interface program.

Once running the application has control of the user's session, in other words the user is in the application program.

The return to HP DeskManager can be made in one of two ways:

- The application (the SON process) can suspend to allow HP Desk (the FATHER process) to perform a task. Once the task is complete, the SON process is resumed, without having to be recreated.

- The SON process can be terminated by the father. Complete control is returned to HP DeskManager. If the user starts up the application again, the SON process has to be recreated.

Ways of suspending the Application

The application can be suspended using Interrupt modes. There are 16 different Interrupt Modes which can be used to suit the application's needs. They can be divided into two main areas, those which:

- Put users into HP DeskManager at the SUSPEND prompt. User's can then move around their desk as normal. They are returned to the application when they press the RESUME key.

- Leave users in the application so that they do not see the HP Desk user interface. In this situation, the application asked HP Desk for some information, or to perform a particular task, for example, display a HELP screen or pass an item to the WORKAREA.

The 16 application interrupt modes are:

HPDESKMANAGER

Application Interrupt Modes

1 Return to HPDESK In Suspend mode
2 Return a file to HPDESK
3 Copy an Item from HPDESK
4 Return a file to the Work Area
5 Copy an Item from the Work Area
6 Replace an Item In the Work Area
7 Initialize Forms Processing
8 Get next forms data
9 Copy a file to HPLIBRARY
10 Document conversion
11 Help
12 Copy from current area Item #
13 Copy from current area subject
14 Check for new mail
15 Execute MPE file as a script
16 Exit

HEWLETT PACKARD
Business Computing Systems

DIXT5

Application Data Passing Examples

Here are some application data passing examples:

1. Send VPRINTFORM File - When in a View based application, we can use the intrinsic VPRINTFORM to send an image of the current displayed view form to an MPE file. This file can then be passed to the users WorkArea and then sent on to other desk users with comments about the attached screen.

2. Controlled Environment - Using Logon UDC's with NOBREAK option and disable MPE for DESK users, we can create an environment where the user will never see an MPE prompt. Every action that the user needs can be done with a script file. Some DESK commands can be deactivated to insure the user cannot accidentally cause damage.

3. Help Page Access - Many applications are lacking in a structured help system. Applications running under HP Desk can now access HELP pages installed in HP Desk. This feature gives the appearance of tight integration and provides a simple solution for those applications where a HELP facility is required.

TASK AUTOMATION

Single user-defined commands for:

- **Automatic mailing and administration of scheduled reports.**
- **Simplified execution of external applications.**
- **Productivity improvements through automated user tasks.**

What are they?

A script file is a normal, named HP DeskManager text item containing a set of commands. When you execute a script, the commands held in it are carried out.

What set of commands can I use in Script Files?

The commands which can be included in scripts can be divided into four main types.

- Normal HP Desk commands, any command you would type while using HP DeskManager for example READ, SEND, LIST, PRINT, PROFILE.
- MPE commands, most MPE commands are allowed in scripts.
- Directives, which control the flow of the script and provide other "programming" type functions.
- Label lines, these allow conditional execution of parts of a script.

There is another feature of script files called active functions. They allow you to use the other types of command in a more sophisticated way providing arithmetic and logical functions.

SCRIPT FILES

* DESK, SCHEDULE and FILE/LIBRARY Commands

* MPE Commands – Can Run an External Application

* Script File Command Language:

  - Arithmetic, String and Logical Functions
  - File and Data Item Handling
  - User Response
  - Parameter Substitution

* Access Controlled by User Groups

Script Files - Example

Here are some examples of what script files can be used for:

1. Personal maintenance function - A script that would scan your filing cabinet and move items that have been created before a certain date to another folder called Old Items. These can then be reviewed and deleted if need be.

2. Report Forwarding Scripts - This script would check to see if a report file exists in MPE, which was created by another JOB. The user would then have the option to send it out for distribution and file it in a Report History folder.

3. Dynamic Info String - Many utility programs and applications will accept info strings at run time. By using script files we can prompt the user for information then format that information into proper info string that can be executed.

Introducing HP DeskManager Intrinsics

History

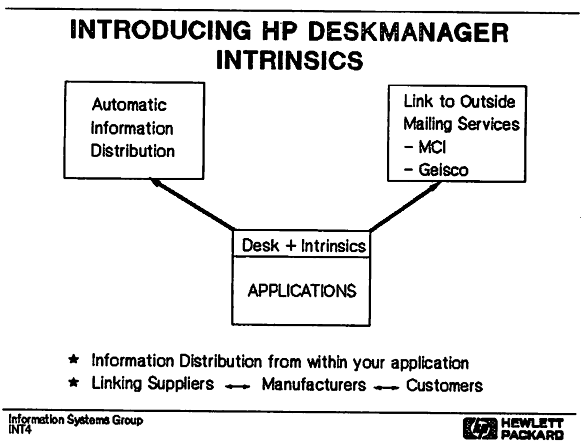
As you have just read HP DeskManager offers extensive integration facilities, from script files through to application data passing. However, our customer base wanted more.

We conducted a survey last year of key accounts and value added businesses (VAB's) and it became clear that there was a need for a programmatic, or intrinsic interface to HP DeskManager.

As a result of this feedback we have developed the HP DeskManager Intrinsics - a new BOLT-ON product for HP DeskManager. (HP DeskManager is also required)

HP DeskManager Intrinsics at a glance

HP DeskManager Intrinsics allow an application to access in a simple way the mailing and other services of HP DeskManager from within that application. The application can automatically distribute information to HP DeskManager users. In addition users can have direct access to HP DeskManager WITHOUT leaving the application they are using at the time. Also applications connecting HP DeskManager to other mailsystems can be more functional and have better performance than is possible with the foreign service connection.



Product Capabilities Indepth

The Intrinsics allow a user to access in a simple way the mailing, diary, directories an converters of HP DeskManager from within his own application programs.

The Intrinsics comprise of two types of access intrinsics with a third set of supporting intrinsics.

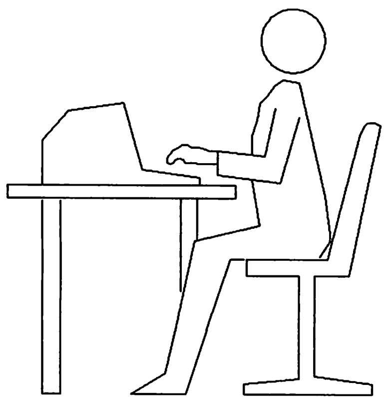
HP DESKMANAGER INTRINSICS PRODUCT OVERVIEW

- ★ User Intrinsics
  - Intrinsic access to HP DeskManager mailing and diary features
  - Allows appliation to sign on as an HP Desk User
  - Allows application integration from outside the HP Desk umbrella (application as father)
- ★ Gateway Intrinsics
  - Gateway to other mailing systems
  - Superior alternative to the foreign service connection
- ★ Support Intrinsics
  - Support for both "user" and "gateway" intrinsics
    e.g. access to Word Processing Converters and the Directory

Information Systems Group
INT5

HEWLETT PACKARD

User Intrinsics



HP DeskManager User

PIC1

These allow an application program to be able to log on as if it were an HP DeskManager user and send and receive messages on behalf of that user via HP DeskManager.

Usage of the User Intrinsics

The User Intrinsics will be used to distribute reports from the application programs to HP DeskManager users, or, conversely, allow HP DeskManager users to supply information to these application programs.

Users of the User Intrinsics are expected to have elementary or advanced applications programming experience and have an understanding of MPE (or MPE/XL) and the MPE file system. Some working experience of the HP DeskManager user interface and knowledge of the local HP DeskManager configuration is required. An understanding of HP DeskManager internals is not required.

User Intrinsics Available:

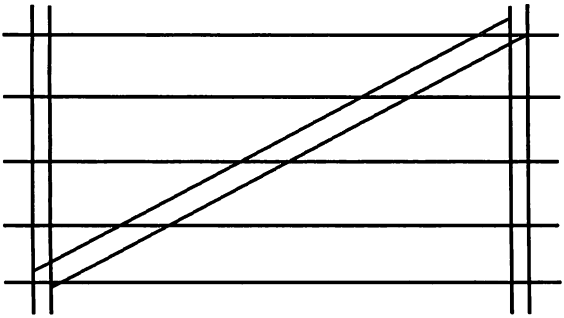
User Signon - allows application to sign onto HP DeskManager like a normal HP DeskManager user.

User Send - allows application to send one at a time, messages or replies to users on the HP DeskManager network.

User Receive - allows application to scroll through the messages in the user intray in sequence, returning both read and unread messages, or it allows the selective extraction of messages. This intrinsic may also be used to return messages in the user pending tray for tracking purposes.

Diary - allows applications to create items in the users diary, to list the contents of the users diary selectively and to extract or delete items selectively.

Gateway Intrinsics



HP DeskManager Gateway

PIC2

These allow a user application program to sign on to an HP DeskManager gateway and send or receive messages on behalf of any user associated with that gateway.

Usage of the Gateway Intrinsics

The Gateway Intrinsics will be used to link mail systems to HP DeskManager instead of using the foreign service connection.

Users of the Gateway Intrinsics are expected to have advanced programming experience and are likely to have some knowledge of HP DeskManager configuration details and issues. In addition they will be familiar with the external applications which will be calling these intrinsics.

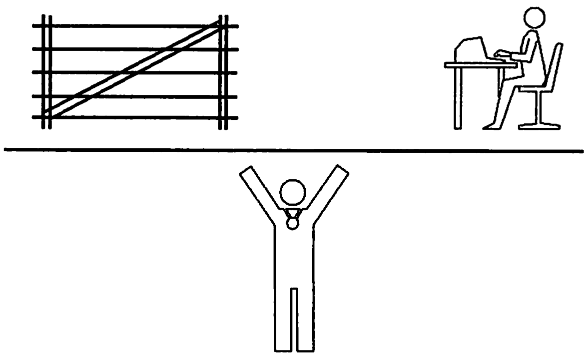
Gateway Intrinsics Available:

Gate Signon - allows application to sign.on to an HP DeskManager gateway.

Gate Send - allows application to send messages, replies and acknowledgments to users on the HP DeskManager network on behalf of any user associated with the gateway.

Gate Receive - allows application to scroll through the messages, replies or acknowledgments deposited on the mail nodes associated with the gateway, returning urgent messages followed by normal th deferred messages.

Supporting Intrinsics



Supporting Intrinsics

These provide support for both the User and Gateway Intrinsics. Services which will be made available to users applications include:

- The ability to look up HP DeskManager directories for a given name.
- The ability to convert a document from one format to another.

Supporting Intrinsics Available:

DeleteMsg - allows application to delete the last message accessed by the User Receive or Gate Receive intrinsics

Name Probe - allows applications to search the HP DeskManager directories to verify one or more names, to extract or convert foreign address information and to extract lists or configured users.

Explain - returns a one line text message relating to the error code resulting from one of the other intrinsic calls.

Utility - allows listing of the intray or pending tray either in full or selectively, and conversion of documents from one format to another (given the converters available on the system).

Terminate - allows application to sign off as the HP DeskManager user.

When should I use scripts and application data passing and when should I use the HP DeskManager Intrinsics?

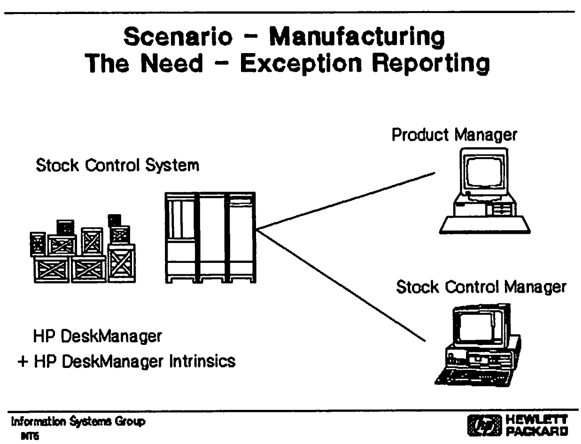
Scripts should be used to automate repetitive tasks currently done manually within the HP DeskManager environment. Intrinsics should be used to automate event driven activities within the application environment.

Application data passing should be used when the principal user interface is HP DeskManager. Intrinsics should be used when HP DeskManager is not the principal user interface and where efficient distribution of information from within an application is required.

Scenario's

The following scenarios illustrate how HP DeskManager Intrinsics have brought profit improvements to different organizations. We have assumed in each case that the organization already has HP DeskManager.

1. HP DESKMANAGER INTRINSICS COST JUSTIFICATION: MANUFACTURING



COMPANY:

Company X - Manufacturing Division.

CONTACTS FOR MORE INFORMATION:

Bill Franklin, Central Office Marketing, Cupertino, 408-447-1156.
Peter Dunmore, Office Productivity Division, Pinewood, England.

BUSINESS PROBLEM:

Stock control clerks waste valuable time reproducing and distributing stock control reports. Valuable production time can be lost because the decision makers in a manufacturing division do not learn about adverse stock levels in a timely fashion. As a result overtime rates need to be paid to production workers to make up for lost production.

NEED DETERMINATION QUESTIONS:

High Level: Do you lose production time because of insufficient levels of stock?
If so how much does that cost per annum?

User Level: How much time does your stock control clerk spend reproducing and distributing stock control reports?
How much does it cost to produce these reports?

SOLUTION:

Integration of your Stock Control Application with HP DeskManager using HP DeskManager Intrinsics.

CUSTOMER SITUATION:

The stock control application generates four stock control reports each month. These reports have to be copied and distributed to, on average, ten people within the organization who need to act upon this information. It is the job of the stock control clerk to perform this function.

SAVINGS:

TOTAL ANNUAL COST SAVINGS FOR THIS COMPANY ARE $23688 PER ANNUM. THIS GIVES AN ROI OF 42% AND THE INVESTMENT HAS A PAYBACK OF LESS THAN 29 MONTHS.

Every month the stock control clerk spends time collating, copying and distributing reports to key individuals in the organization. This costs on average $154/month. These reports, generated by the manufacturing application, indicate shortages or excessive stock levels. This information is important to ensure efficient stock management and to keep the production lines running. What the organization has found is that key information on stock levels has not been made available quickly enough for the necessary corrective action to be taken to prevent a loss in production. To make up for this loss in production, overtime rates have to be paid.

Using HP DeskManager and the HP DeskManager Intrinsics the exception reports are distributed automatically to those who need the information. As a result the organization has seen a 30% decrease in overtime manufacture. With overtime payments running at 10% of operating wages, ten workers effected and labor rates of $35 /day, the organization has seen a cost displacement due to reduced overtime payments of $1820 /month.

KEY ASSUMPTIONS:

Labor rate per hour for a Non-Professional ($35.00)	______
Labor rate per hour for a Professional ($65.00)	______

INVESTMENT:

27562A - HP DeskManager Intrinsics	$0
27562A - Opt. 330 first copy for S/70 HP3000	$9000.00
Three man months programming (65x8x90)	$46800.00
TOTAL INVESTMENT	$55800.00

WORKSHEET

A. Labor savings to Stock Control clerks copying and distributing stock control reports as a result of the integration with HP DeskManager using HP DeskManager Intrinsics

Situation 1:

Number of reports produced per month (4)	x	______
Number of copies to be made per month (10)	x	______
Time spent copying each report (1 min)	x	______
TOTAL time for preparing reports / month (40 mins)	=	______
Labor rate for Clerk $35 / hour ($0.58/min)	x	______
TOTAL cost for A ($23.20/month)	=	______

B. Cost savings due to production and distribution costs of the stock control reports.

Situation 2: Production Costs

Number of reports produced per month (4)	x	______
Number of copies to be made per month (10)	x	______
Number of pages per report (10)	x	______

Cost per page (0.02 x 1.714)	x	______
TOTAL cost of producing report/month ($13.7)	=	______

Situation 3: Distribution Cost (local)

Number of reports produced per month (4)	x	______
Number locally distributed (8)	x	______
Time to distribute the reports (5 mins)	x	______
Labor rate for Clerk $35 / hour ($0.58/min)	x	______
TOTAL cost of local distribution/month ($93.3)	=	______

Situation 4: Distribution Cost (Fax)

Number of reports produced per month (4)	x	______
Number faxed to destination (2)	x	______
Cost to fax per report (1.7x1.714)	x	______
TOTAL cost of faxing out reports/month ($23.31)	=	______

Total cost per month A+B ($154)	=	______

C. Cost Displacement due to reduced overtime rates

Situation 5:

Decrease in overtime manufacture (30%)	x	______
Overtime as a % of operating wages (10%)	x	______
Number of production workers effected (10)	x	______
Hours in day/Days in week/Weeks in year(8x5x52)	x	______
Labor rate for Production Worker $35 / hour	x	______
TOTAL opportunity cost / month ($1820)	=	______

TOTAL cost displacement due to integration with HP DeskManager Intrinsics / month A+B+C ($1974)	=	______
TOTAl cost displacement per annum ($23688)	=	______

WORKSHEET SUMMARY

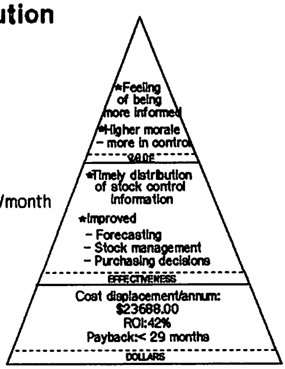
Total Cost Displacement Savings ($23688)	______
Return on Investment (23688/55800=42%)	______
Time to Payback (Less than 29 months)	______

Scenarios – Manufacturing
The Solution

★ Investment
- HP DeskManager Intrinsics
"A" copy $9000
- 3 man months programming $46800

★ Cost Displacement
- Stock control desks time/report production and distribution costs $154/month
- Decrease in manufacturing overtime $1820/month

★ Effectiveness
- Timely distribution of exception reports leads to minimum production loss due to stock shortages



Information Systems Group
IXT7

HEWLETT PACKARD

2. HP DESKMANAGER INTRINSICS COST JUSTIFICATION: MSS

COMPANY:

Company X - Marketing Sales and Service.

CONTACTS FOR MORE INFORMATION:

Bill Franklin, Central Office Marketing, Cupertino, 408-447-1156.
Peter Dunmore, Office Productivity Division, Pinewood, England.

BUSINESS PROBLEM:

The sales organization needs to be informed of the status of their customers orders on a very regular basis in order to ensure customer satisfaction. The supervisor of the order processing clerks needs to be able to more effectively manage the wordload of his direct reports in order that they can cease to waste valuable time producing and distributing this information to the sales organization. The sales people spend a noticeable amount o their time attempting to find out the current status of their customers orders.

NEED DETERMINATION QUESTIONS:

High Level: Does your sales force need regular and automatically distributed information about their customers order status?

User Level: How much time does your order processing clerk spend reproducing and distributing order status information?
How much does it cost to produce an distribute this information?
How much time do your sales force spend attempting to find out the current status of their customers orders?

SOLUTION:

Integration of your Sales Order Application with HP DeskManager using HP DeskManager Intrinsics.

CUSTOMER SITUATION:

ToDo lists are generated within HP DeskManager (using the Diary Intrinsics) and displayed within the Sales Order application. The supervisor automatically receives a consolidated ToDo list from all of the sales order clerks and can re-assign work where appropriate. In addition, information, such as acknowledgments, can be automatically sent out to the field and the factory so that they have the most up-to-date details on their customers order status. The current method involves the salesman, or whoever it is who needs the information, to ring the office and asking the sales order clerk for the information, who then spends time producing and distributing it.

SAVINGS:

TOTAL ANNUAL COST SAVINGS FOR THIS COMPANY ARE $215200.80 PER ANNUM. THIS GIVES AN ROI OF 200% AND THE INVESTMENT HAS A PAYBACK PERIOD OF LESS THAN 6 MONTHS.

Each time an order is logged a series of events need to occur so order that the correct set of acknowledgments and confirmations are distributed to the customer, the field and the factory. It is the job of the sales order clerk to monitor the activities following an order being logged, and to relay this information to all the people concerned. The salesmen themselves need to know the current status of their customers orders to ensure that the order is progressing as it should and the customer remains a satisfied one. To obtain this information they make regular calls to the sales order clerk. It is estimated that they would regain 1% of their time to pursue more productive tasks if this process were automated and they received the information they needed automatically. This results in a cost displacement of $17933.40 per month. It is estimated that the sales order clerk makes 2 calls per order per salesman, in order to find out the current order status. With the integration of HP DeskManager and the sales order application using the Intrinsics this function happens automatically and results in a saving of $1166.77 per month. The sales order clerk then has to produce and distribute the information which takes on average 3 minutes per order per salesman. Here a saving of $1166.77 per month is made because the sales order clerk is now free to perform other tasks.

In total, a saving of $17933.40 per month is made, due to the integration of the sales order application with HP Deskmanager using HP DeskManager Intrinsics.

KEY ASSUMPTIONS:

Labor rate per hour for a Non-Professional ($35.00)	______
Labor rate per hour for a Professional ($65.00)	______
Number of Salesmen (100)	______
Average number of orders/salesman/month (10)	______
Average size of order ($4000.00)	______

INVESTMENT:

27562A - HP DeskManager Intrinsics	$0
27562A - Opt. 320 first copy for S/37 HP3000	$5000.00
27562R - Opt. 320 right to copy for S/37 (2)	$8500.00
Six man months programming (6x30x65x8)	$93600.00
TOTAL INVESTMENT	$107100.00

WORKSHEET

A. Labor savings to Order Processing clerks.

Situation 1: Personnel cost in ringing the factory for information:

Average number of orders per month (10)	x	______
Number of Salesmen (100)	x	______
Average number of calls to factory/order (2)	x	______
Average length of call (1 min)	x	______
TOTAL time for calls to factory / month (2000 mins)	=	______
Labor rate for Clerk $35 / hour ($0.58/min)	x	______
TOTAL cost for situation 1 ($1166.67/month)	=	______

Situation 2: Personnel cost in producing and distributing the information to the salesmen.

Average number of orders per month (10)	x	______
Number of Salesmen (100)	x	______
Average time to produce information/order (2 mins)	x	______
Average time to distribute information/order (1 min)	x	______
TOTAL time to produce/distr. / month (2000 mins)	=	______
Labor rate for Clerk $35 / hour ($0.58/min)	x	______
TOTAL cost for situation 2 ($1166.67/month)	=	______

TOTAL costs for A ($2333.34) = ______

B. Labor savings due to the sales force having more time to focus on productive tasks.

Situation 3:

% of time saved per annum (1%) x ______
Number of Salesmen (100) x ______
Labor rate for Sales Profess./annum (65x8x30x12) x ______
Number of months in the year (12) / ______
TOTAL savings / month ($15600.00) = ______

TOTAL cost displacement due to integration with HP DeskManager Intrinsics / month A+B ($17933.32) = ______

TOTAL cost displacement per annum ($215200.08) = ______

WORKSHEET SUMMARY

Total Cost Displacement Savings ($215200.80) ______
Return on Investment (215200/107100=201%) ______
Time to Payback (Less than 6 months) ______

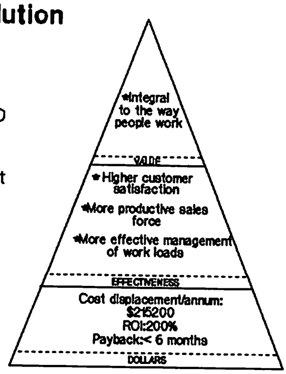
2. Marketing, Sales and Service

Scenarios – Marketing, Sales and Service
The Solution

★ Investment
- HP DeskManager Intrinsics "A" copy and 2 "R" copies $13500
- 6 man months programming $93600

★ Cost Displacement
- Order Processing Clerks time/report production and distribution costs $2333/month
- Salesmans time $15600/month

★ Effectiveness
- Greater customer satisfaction



Information Systems Group
INT7

HEWLETT PACKARD

3. HP DESKMANAGER INTRINSICS COST JUSTIFICATION: GATEWAY INTRINSICS

COMPANY:

Company X - Link to a Proprietary Mailing System

CONTACTS FOR MORE INFORMATION:

Bill Franklin, Central Office Marketing, Cupertino, 408-447-1156.
Peter Dunmore, Office Productivity Division, Pinewood, England.

BUSINESS PROBLEM:

The organization needs to have an efficient, high functionality link between HP DeskManager and it's own proprietary mailing system.

NEED DETERMINATION QUESTIONS:

High Level: Do you have your own proprietary mail system and HP DeskManager running side by side with no link between the two?

User Level: Have you considered making this link using the Foreign Service Connection available within HP DeskManager?

SOLUTION:

Use the GATEWAY HP DeskManager Intrinsics as a means to integrate HP DeskManager with your Proprietary mailing system. Using the Gateway Intrinsics provides improved performance (approximately 30%), greater functionality (e.g. two way acknowledgments), and easier administration in comparison with the Foreign Service Connection.

SAVINGS:

TOTAL ANNUAL COST SAVINGS FOR THIS COMPANY ARE $46800.00 PER ANNUM. THIS GIVES AN ROI OF 936% AND THE INVESTMENT HAS A PAYBACK PERIOD OF LESS THAN 2 MONTHS.

KEY ASSUMPTIONS:

Labor rate per hour for a Professional ($65.00) ______

INVESTMENT:

27562A - HP DeskManager Intrinsics	$0
27562A - Opt. 320 first copy for S/37 HP3000	$5000.00

TOTAL INVESTMENT $5000.00

WORKSHEET

A. Programming time to link HP DeskManager to the proprietary mailing system versus the Foreign Service Connection:

12 man months programming using FSC
(12monthsx30daysx8hoursx$65/hour=$187200.00)
9 man months programming using HP DeskManager Gateway
Intrinsics (9mthsx30daysx8hoursx$65/hour=$140400.00) -
Saving due to use of Intrinsics ($46800.00) =

TOTAL cost displacement due to integration with HP DeskManager
Gateway Intrinsics ($46800.00) =

WORKSHEET SUMMARY

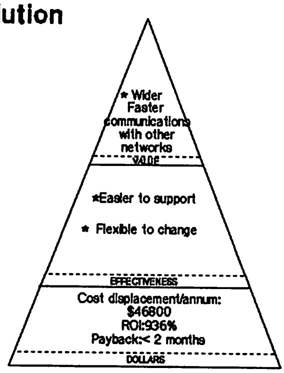
Total Cost Savings ($46800.00)
Return on Investment ($46800/$5000=936%)
Time to Payback (Less than 2 months)

Scenarios: Gateway Intrinsics
The Solution

* Investment
  - HP DeskManager Intrinsics
  "A" copy $5000

* Cost Displacement
  - 25% less programming time
  (ie. 3 man months) $46800

* Effectiveness
  - Faster communication with other networks with a product which is easier to support and change



Information Systems Group
INT7

HEWLETT PACKARD

Conclusion

This paper talked about the integration features available with HP DeskManager.

The paper introduced HP DeskManager Intrinsics (a new BOLT-ON product to HP DeskManager) and showed via examples how organizations could gain profit improvements from using the product to integrate HP DeskManager with other applications.

HP DESKMANAGER INTRINSICS
Summary

* Information Distribution provides proven profit
* HP DeskManager Intrinsics provides Information Distribution from within your application
* CPL - 3rd Qtr 1988
* Availability - 3rd Qtr 1988
* Pricing - $2000 to $11000 depending on processor

Information Systems Group
INT10

HEWLETT PACKARD

Business Information on the Desktop:
Alternatives to Paper-Based Reporting

This paper outlines how information technology can provide a competitive advantage and is based largely upon the theories of Michael Porter of Harvard Business School. Porter said that rapid technological changes in information systems is having a profound impact on competition and on competitive advantage. I would like to begin by showing you why we think information technology can have strategic significance and then move into a brief explanation of how it changes the nature of competition.

We feel that information technology has strategic advantages for two reasons. The first reason is that it transforms what Porter calls the value chain. It can actually alter the activities that a company performs in a couple of ways. Porter defines the value chain as all the economical and technological processes or activities that a firm undertakes to produce its product, and these activities are all related to one another in what we call a value chain. Think of the product from a concept until the product goes out the door (whether it is a tangible product or a service) as the sum total of the number of incremental steps in the entire process. Each of these distinct activities has its own cost driver or its own associated processes with costs. One of the ways to understand what a company does is to look at all those individual steps. Within those steps, two kinds of processes occur. One is information processing and the other is physical processing. By information processing I mean individual steps of data collection, manipulation and communication of information.

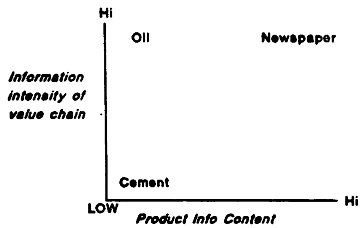
For example, an MPR system is a mechnism for tracking the information processing portion of producing a product. Physical processing is simply the manufacturing of the product. Information technology has entered physical processing in products like machine tools which are now operated with computer controls. In summary, information technology has entered the workplace along that value chain in a number of different areas during the process, one of which is information processing and the other is in the physical processing.

The other area that has provided significant strategic advantage is where value chains can be linked. For example, McKesson Drugs has put terminals in their drugstores right on the pharmacists' desks. The pharmacist can check McKessons inventories, in fact check their own orders against McKesson. It used to be that the sales rep would visit once a month or every two months, now McKesson Drugs has actually provided the pharmacist with on-line information. What they've done is linked their activities together and in this way we find that information technology is providing strategic significance, in fact transforming the way businesses operate.

To summarize we really see the value chain being transformed. We see it being transformed in terms of information processing, physical processing, and the linkages between value chains of different organizations.

The other way that information technology is providing strategic advantage is that it is actually transforming the products themselves, not just the processes but the actual products. An example might be a product like Compustat which is a product that could not have existed several years ago because the information technologies did not provide the distribution mechanism. Another example would be be my dishwasher which came with a micro chip in it and now it has enough information bundled with it to tell the difference between pots and pans and china. Electronic fuel injection in automobiles is another example. Information is transforming whole products.

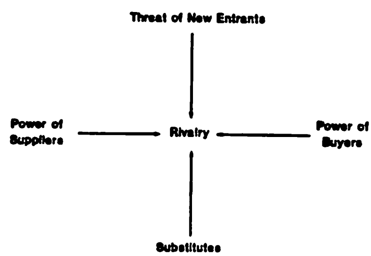
So we have two significant things happening --the whole value chain and the way companies are building their businesses is changing, and we have products that are changing. Two key trends emerge from these facts. The information content of products is increasing and the information content of the value chain is rising, providing more and more opportunities to insert into this process information processing. See diagram below for examples:



A product for which the processing (either service or tangible product) is very complicated and involves a lot of information is oil. The refining, locating, and delivery of oil is very sophisticated. For example, the information technologies that are being applied to squeeze more oil out of the ground and so forth. But, the actual information content of oil is very low so it falls on the left side of the scale. Cement on the other hand, is an example of a product that has low information processing content and also low information within the product itself. Cement, like oil, doesn't have any information associated with it but the processing of it is relatively simple - take a shovel of cement, 3 shovels of sand , add a little water and that's all there is to it. But note the quadrant containing newspapers. Here you have a product that has a lot of information and also the processing of it is very complicated. For example, USA Today has 17 distribution outlets and they transmit, electronically, information to each of those sites to publish the newspaper. The process to get your USA today is very complicated as well as the product itself having a lot of information.

To highlight two key trends, the value chain or the processes that companies use to produce their products are more and more beginning to have more information content and also the products today also are beginning to have more information built into them. I feel these are the reasons why information technology can have strategic benefit, because it can transform products in these two areas.

Let's briefly look at how information technology can change competition and the structure of an industry. Relying on Porter's Five Factor Analysis of an Industry, the forces listed on the following diagram all act together to determine an industry's structure.



The following examples demonstrate how information can disrupt an industry that has been traditionally stable. Some examples in each of the five areas:

The Power of Buyers - We have seen some examples of this with Videotech's Home Teleshop where you can actually order products on your television. This has changed the whole picture for the distribution channel in the department store. Another example would be on-line mortgage services when you buy a house and can get information on-line about potential mortgages that are available. The last example might be the airline reservations system used by many airlines including United and American which provides on-line information to travel agents on all their flights, and this has really changed the structure of the industry. Suddenly consumers have much more information about buying and as a result more bargaining power. Technology here has disrupted the industry in some fashion.

The Threat of Substitution - A good example here is LEXIS an on-line database product that provides database access to legal information. Large volumes of legal information are put on ROM discs and provide key word search to alter the way in which legal information is delivered. By putting information on discs, they are in a sense becoming substitutes to the traditional legal publishers. So, again, technology changing the nature of the business.

Bargaining Power of Suppliers - A good example here is again McKesson where a supplier has locked up a distribution channel, the pharmacists, by providing a terminal right in the pharmacy. They have created switching costs so that if the drugstore wanted to change suppliers they would have to locate someone else, change terminals, hook the new suppliers equipment up and would likely encounter some problems in terms of their transition.

Threat of Entry - The best example here might be Merrill-Lynch's Cash Management Account. Merrill-Lynch has developed entry barriers by instituting a cash management account where you actually call up and receive a complete rundown of you account. They have errected barriers for other corporations that want to have cash management accounts because of the massive technologies in place in order to provide that on-line service for the account managers.

Rivalry Among Competitors - One of the best areas to look at this is in the distribution businesses. Distributors have implemented technologies to achieve more rapid and accurate distribution, tieing themselves to suppliers, and being able to speed-up deliveries. In fact some trucking companies have provided on-line services that allow firms that are expecting a delivery to locate their shipment. As a result, these distribution houses have developed high fixed costs. To cover these costs, businesses have expanded markets and intensified competition and rivalry.

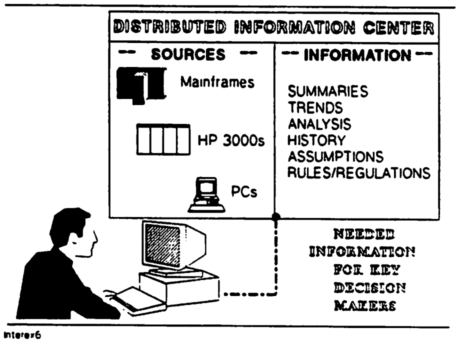
So as you can see, technology can affect industry in a number of different ways and we believe that technology can have significant strategic advantage. Porter summarizes this by saying that there are three key ways you can create competitive advantage in your business. The first one is to reduce costs. An excellent example of this is Caesars Place, where they have instituted an information system which tracks who their high rollers are. They are able to provide services such as free rooms, complimentary dinners and so forth to just those people whom they know are high rollers. So with information systems tracking, they are able to reduce their costs of supplying high rollers with the services they require and at the same time build their business.

The second key area that Porter points out is differentiation. Information technology can help you differentiate your product. The Merrill-Lynch Cash Managament Account and American Express Travelers Services where American Express has targeted specific business travelers and provided new services for them are two good examples of differentiation.

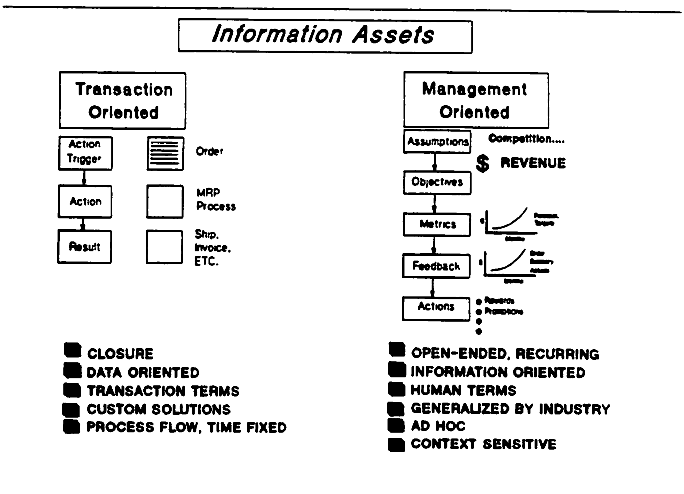
The third area is that information technology can actually spawn whole new businesses. Here, USA Today is a good example; you could not have a national newspaper until you have the capability of distributing information electronically.

What does this mean for you? If you want to analyze your industry to determine whether you can develop advantages through technology , look at just a couple of things. The information intensity of the industry, both in processes and in products, and the trends associated with the information content of your product. If you can assess that and determine that there is a high potential for information technologies to be applied to your business and then apply them appropriately, you can achieve competitive advantage. Do this five factor analysis to determine whether there is potential in your industry for structural change. Identify and rank the potential areas for you to acheive strategic advantage in your industry and then develop an action plan and implement it. With this as background I'd like to show how OSD Marketing institiuted strategic advantage through Information Access by developing an information center.

We define distributed information center as the pooling of information such that the information becomes more than just data.



It has context, it has meaning, it has relationship, it provides information for decisions so it goes much further than just meaningless datapoints. With that as background, let's explore some of the trends and the history of these information centers.

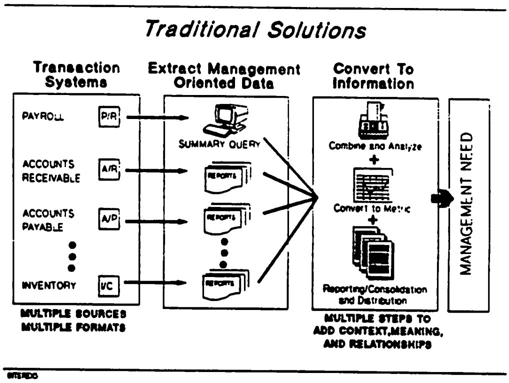


One of the key trends is the understanding of information as an asset. As an asset when information becomes pooled you can then adopt the objective of utilizing that asset to forward your business. Another trend is that the MIS manager once he has pulled this information can have strategic influence as a line manager within the business. No longer just a DP shop, now the MIS manager becomes important strategically for the firm in order to achieve the kind of advantage that Porter is talking about. Another key trend would be the recognition of the difference between transaction-oriented data and management-oriented data. Transaction-oriented data is the traditional data processing machine. It is a process that has an action trigger like an order coming in and has a series of actions such as picking list is filled out, the order is picked from the shelves and packaged and shipped, and then it has a closure or loop or which is the result which is invoicing and payment by the customer These traditional systems have had closure and are contrasted to what we are beginning to see as the emergence of the information center. These centers are open-ended and the processes that are involved in them are recurring and the data is being used by managers and doesn't have the action, trigger, results effect; it tends to be more open-ended. So the actions that are employed in information centers are on-going, dynamic, flexible, ad-hoc as we will point out later.

The data in transaction-based systems is usually coded. It has a number of digits and you don't know exactly what that data means. In other words, it doesn't have context, whereas, when you look at management-oriented systems (information centers), the data becomes informational - rather than a code 15 it's sales region, southern. Or, rather than A or B you know that it is the salary file. The data is expressed in human terms, understandable terms.

Another thing that has happened is that the solutions within within that value chain are less customized from a management perspective. Management has a higher level view of exactly what a company is doing, they need income statements and balance sheets. They don't have to worry about the individual processing that is occurring in each step of the value chain so rather than very customized solutions like a transaction-based system, they are generalized. Management-oriented systems have real ad-hoc ability, they are context sensitive So what has happened traditionally is we have moved away from these transaction oriented systems towards management-oriented systems. It has been slow, but it has been the foundation of the information center.

The graph below illustrates a traditional solution.



On the left, the traditional transaction system, closed loop processes which are just data and the tasks associated with trying to get that information to meet a management need are on the right. These generally involve multiple steps: it is reports, query screens, review screens, etc. and it involves the manual rekeying of data into Lotus spreadsheets. You have to combine and analyze it to get it in terms you can use to make management decisions. These multiple steps resulted in a long and laborious task. What we are saying today today is that Information Access has provided the solution to that long task and provides management-orienated information which is pulled together in a contextual relationship such that you have a rational method of collecting, managing, and distributing information so that the information can be used to make decisions, expedite action and achieve competitive advantage. The information center becomes the first step in implementing a system by which you can support and achieve competitive advantage once it has been isolated within the industry.

- Management Oriented -

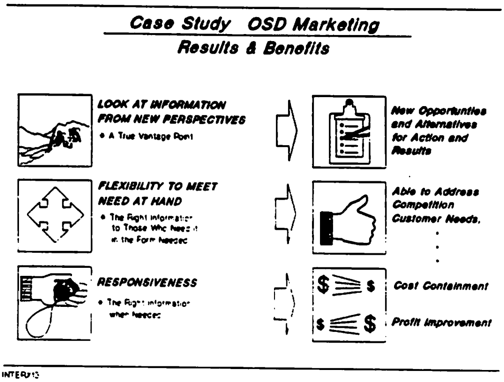
THE INFORMATION NEED

- ***A rational method of collecting, managing, and distributing information which gives us a clear focus to:***
  - ***Make better decisions***
  - ***Expedite action***
  - ***Maintain quality***
  - ***Evaluate results***

On a company wide, sub-entity, workgroup and individual basis

---

The implementation of this Information Center has provided a way for us in HP Product Marketing to look at our information from a totally different perspective. For the first time, because the Information Center has given context and relationship to the data, we know that for instance, products are selling stronger in one region than in another. Or, we understand that discounting structures are causing discounts to occur at higher rates in one part of the country or the world than another, so we begin to get a whole new look at the data.



This ties into one of the three key things that Porter points out as ways of developing competitive advantage. It offers the opportunity to spawn new businesses or look at new ways of doing your business. You may have had an opportunity, for instance, to implement promotions based upon this new information you receive from this information center.

The second point is the flexibility of the information, the adhoc, dynamic, on-the-fly use of Information Access provides a way to differeniate our product. We can get the right information when we need it on a timely basis and if it's getting close to the year-end and we are under sales, we can execute a program to improve the situation. It enables us to look at our customers and see what type of customer is buying and how to differeniate the product.

For example, if we find that we are getting more sales in a specific segment, we can understand that and begin to better meet customers needs. That is Porter's second point, differeniate your product with information technology.

The last point is responsiveness - the timeliness of the information. A good example is a team of accountants who are using a Lotus spreadsheet format for summary reports. From an actual beta site, we found that by using Information Access to automatically link the data into your Lotus spreadsheets, we have been able to increase productivity from a 12 hour process to an hour and a half process. This really cut the overhead costs! So again, here is an operational application of the technology that provided a cost benefit. With all these things wrapped together we have implemented an information center that provides competitive advantage and allows us a different way to look at our business, achieve results and develop strategic opportunities.

UTILIZING DATA PROCESSING INFORMATION
TO SUPPORT MANAGEMENT DECISIONS

I) Information Technology = Competitive Advantage

A. Why does information technology have strategic significance?

1. Transforms the way value activities are performed.

* information processing component
  dfn: steps needed to capture, manipulate and and channel data to perform task (MRP), substitutes machines for human effort, ledgers give way to computers.

* physical processing component
  dfn: physical tasks required, computer controlled machine tools.

* linkage of buyers and supplies
  McKesson Drug
  JIT inventory

2. Transforming Products

* information component expanding
  LaserRom
  electronic fuel injection
  Compustat, SEC data

Value Chain Technological and economically distinct activities. Cost drivers for each. If total cost of value activities < price then a profit results.

Trends Information content of products is increasing
Information content of value chains is increasing

B. Changes the Nature of Competition

1. Changing Industry Structure

- power of buyers
  * Videotex home shopping
  * on-line mortgage services
  * American Airlines System Sabre

- substitution
  * LEXIS

- entry barriers

*Merrill Lynch, cash management account information

- rivaly
  * distributors increase fixed costs and displaced people, an increase in volume must result

- suppliers
  * tie McKesson to drug stores with in house systems (switching costs)

2. Creates Competitive Advantage

   - lowers costs
     * Ceasors Place

   - differentiation
     * American Express Travel Services for custom products

3. Spawns new business

   - Federal Express Zapmail
   - USA Today

C. Competing in the Age of Information

1. Assess information intensity of product and value chain

```
               Hi |  Oil           Newspaper
Information       |
intensity         |
of value          |  Cement
chain          Lo |______________________________
                 Lo                              Hi
                       Product Info. Content
```

2. Potential for Industry Structural Changes

   - 5 factor analysis

3. Identify and rank ways information technology can provide competitive advantage or new business and implement.

II). The Information Center Technology can achieve competitive advantage

A. Definition of Information Center

- pooled, logically related data with context and meaning

B. Trends

- information is asset

- information is more than data, has context

- MIS has strategic roll

- software technology (I/A) allows information center to come out of transaction database.

C. Emergence of Information Center

- transaction vs management oriented data

- traditional solution

- what is needed!

D. Implementation Example

E. Results and Benefits - How Information Center has yielded New Business, Differentiation or Low cost

- New perspectives

- Flexibility

- Responsiveness

HP Help in Customizing Software

Andy Watts
Hewlett-Packard Ltd.
Pinewood,
England.

Summary.

Hewlett Packards Office Productivity Division, located in Pinewood, England, is one of a number of HP Divisions to provide a *Special Projects* capability. Special Projects are custom software modifications to standard products to tailor them specifically to an individual companies needs.

This paper looks at the approaches and processes that are required to provide an effective Software Customization capability. It shows the many common threads between customized software development and standard product development and also the areas of significant difference.

This paper will illustrate the factors that make projects good candidates for specials and what makes a bad candidate. It will also show that attention to Quality and Support and the adherence to an effective and understood process for software development is as critical, if not more so, for customized software as it is for standard product development.

The Formation of a Specials Capability

Before describing the mechanism of special projects, it is worth looking back at the major forces which make the availability of such a service necessary.

Over the last 5 to 10 years the Office Automation industry, vendors and customers alike, has undergone a major shift of emphasis. Initially, efforts where concentrated on standalone problem solving capabilities such as word processing and spreadsheets. Recently, the requirement to provide these facilities in an integrated networked environment has become the driving force.

As customers and vendors have become more and more sophisticated in the systems that they build, these systems are becoming mirrors of their organizations. While some companies control their Office Systems from one central location, others have decentralized administration and purchasing policies. While some companies, like HP, use electronic mail as an extension of an 'open door' communications policy, others, such as the military, use it's capabilities to ensure that proper communications channels and security procedures are adhered to. In addition for many companies, their Office Systems, or more correctly their implementation of Office Systems, are their competitive edge.

Increasingly these factors are leading companies to recognize the need to tailor Office Systems for their own individual organizational needs. HPs Office Systems, and those of other major vendors, incorporate a comprehensive suite of 'customization' tools as part of the standard product. And indeed, organizations exist, both internal and external to the vendors, that perform customization work using these facilities.

However, there will always be a more specific and specialized need. There will always be features, performance and behavior necessary to a companies operation of Office Systems, the provision of which requires detailed knowledge of and access to the internal workings of the products themselves.

There are four ways in which this need manifests itself among the major corporations.

1) Leading Edge Projects.
   Often larger companies, in their continuing effort to maintain and improve their competitive edge, embark on projects which have requirements not yet addressed by the product offerings of the major vendors. These companies require help from their vendors in adding the needed capabilities to the products well in advance of when the vendor would be able to release a generic capability.

2) Trailing Edge Projects.
   In contrast, other companies due either to the logistics associated with implementing large Office Systems in their organizations or from a conservative culture, choose to implement those products which are, for the most part, 'functionally stable'. While it would be impractical for a vendor to provide a permanent enhancement process on older versions of all products, for some companies it is an essential and cost-effective requirement.

3) Utility / Tool provision.
   In the larger office systems products, a wealth of 'unsupported utilities' and administration tools grow up around the product. These originate from a wide range of sources; from the development labs, from the vendors support organization, from the vendors own internal implementation of the products, from third parties and from customers themselves. Some of these utilities are of sufficient general use to be marketed as part of the product line by the vendor. Others are specialized and would not justify the costs of 'productization'. Nonetheless, Customers require access to and, more importantly, support of, such utilities as will improve the efficiency of their implementation.

4) Data Structure Access.
   Often, companies require, in order to develop their own applications, access to unpublished data structures and file formats used by the vendors products. Software vendors are reluctant to publish internal data structures for a number of reasons; It establishes a dependence that prohibits future changes and

enhancements and support of direct access to data structures presents a complex challenge. In order to be successful in their own implementation, however, customers require a mechanism to get access to the data they need, in a supported manner.

These factors, catalyzed by a number of specific projects, resulted in Hewlett Packards Office Productivity Division setting up a Special Projects capability similar to that provided by a number of HPs hardware divisions.

Objectives, Charter and Non-Objectives.

It is important in any service to early on, establish the ground rules. With clear objectives and charter it is easier for customers and HP to determine when a Special project is warranted, feasible and cost-effective.

The Special Projects Charter can be expressed as:

"To provide a Service to HP's customers to enhance and extend OPD products to meet their specific business needs."

"To provide a service ...."
Through Special Projects, Customers can commission enhancements to OPD products. They can obtain those enhancements that only they require and can obtain them in a time frame that is appropriate for their business needs.

"... to enhance and extend OPD products ...."
This is an extremely important aspect of Special Projects. Specials are almost always compatible with the general product direction of OPD. Special Projects are, in the main, incorporated into the standard products at the earliest possible opportunity, providing the customer with the same long term support that they would expect from any HP product.

".. to meet their specific ..."
Special projects are undertaken in partnership with a *single* customer and reflect their specific requirements.

" ... business needs"
Special Projects must *always* provide the customer with a significant return on investment.

It is worth noting here that the objective of ensuring customer satisfaction in HP's products is not an objective of specials. Customer Satisfaction is central to HP's product development philosophy. The responsibility for ensuring that satisfaction remains firmly vested in the product development teams.

What Sort of Projects are Good Specials?

Given the creating forces and objectives, it is possible to identify the sort of projects which are *good* specials. This is best done by a few examples.

Case 1 - HPDesk Addressing Mechanisms

The philosophy of HPDeskManager is one of addressing electronic mail users by name rather than by numeric identifiers. Some organizations address business communications to the job function rather than the individual. This ensures that personnel changes do not need to be reflected in the mail directory. A project to enhance HPDeskManager to accept numeric job code identifiers as user names is a good special.

- o It addresses an individual customer need rather than a generic one.
- o It does not conflict with the general product direction
- o Being a small project, it has sufficient return on investment to warrant a special.

Case 2 - Multi-Vendor Electronic Mail Directory.

A large electronic mail network that encompasses office systems from a number of vendors poses a problem in directory maintenance. Each vendor employs different directory mechanisms. While the issue of standardized electronic mail communications is now addressed by the CCITT X.400 standards, the directory services standard (X.500) is some way off a practical multi-vendor implementation. Customers who currently operate multi-vendor networks need to provide a directory linkage on current technology. This makes a good special because:

- o It requires specialist knowledge of the product to open the directory architecture.
- o It is specific to an individual customer in that the mix of mail systems and architecture of the distributed directory would be unique.
- o Provision of automated Directory services is in keeping with the general product direction.
- o While being quite a large project, the impact on a large network more than justifies the cost.

Case 3 - Product Specification Extension.

Occasionally, customers find themselves using a product in an environment outside the product's specifications. Whereas in hardware terms this might mean using a product which has been certified to -20°C, in an environment where temperatures drop to -70°C. In software terms this might be requiring a throughput that is an order of magnitude in excess of the product's specification or extending a products specification to include support for a specific printer or terminal not previously supported in the standard product. These are appropriate specials because :

- o They require performance or specification in a product beyond that required by other customers.
- o It doesn't conflict with the product or have an adverse affect on other customers implementations.
- o It generates significant return on investment for the customer in terms of either increased throughput or reduced hardware resource.

Projects that are not good specials.

Conversely, there are some projects that are not best handled as specials even though they might initially appear to be.

Case 1 - Low investment return.

A project was investigated to provide a support in HPWord for a previously unsupported printer. In this particular instance, the customer concerned possessed only a very few of these printers. This project turned out not to be a good special. It was a better business decision for the customer to replace the small number of printers with newer models (which they intended to do in the future anyway) than to finance a special.

Case 2 - Not in Line with Product Strategy.

A request to provide, within HPAdvanceMail, a conversion between two third party document converters is a common request. While this is an important requirement for the customer, it does not make a good special project. Support for these third party word processors could never become part of the standard product. A better special project in this case is for OPD Specials to develop a generic converter plug-in mechanism. Then the converter itself could be developed by either customer or word processor vendor or third party or HP Project Center.

Case 3 - Product Dissatisfaction.

A limitation in HPDeskManager's distribution list handling made it difficult to correctly address a user whose name clashed with more than 500 other names. While this limitation is only of issue in very large networks, it does not constitute a special project. Essentially, since this restriction is not a publicized limitation of the product it constitutes a product defect and therefore a solution to this request is the responsibility of the product development teams.

Case 4 - Published Mechanisms

The development of gateway connections between HPDeskManager and other customers own private mail systems is often the subject of specials requests. In most cases these do not require special project developments since the mechanisms to develop gateways are provided through standard HPDeskManager product features such as Foreign Service and the HPDesk intrinsics. This sort of development is normally better tackled by customers own development staff, a local software house or HP's own Project Center Operations.

A Process for Specials.

In order to have a successful project, a thorough process is required.

In all Special Projects, as indeed in all software development projects, the development of a comprehensive Functional Specification, of the problem and proposed solution, is the most important phase. Such a document can help to guarantee success and lower costs of the implementation by:

* Providing a firm project description and plan.
* Ensuring user acceptance and support
* Minimizing changes and delays
* Enabling productive scheduling of people and other resources.

The Functional Specification is most important for it defines and bounds the problems. This specification determines the total system design from a level above the detail level. Additionally determinations are made of the capability to accomplish the desired end. It is also typically utilized to determine the cost of producing the complete system.

In addition a Functional Specification enables accurate projection of project cost, eliminating the need to add a "risk factor".

Special Projects, like any software development activity, have a Life Cycle. It is important to the success of a Special Project for this life cycle to be clearly understood and agreed to by all parties before the project begins. It ensures that the project is managed efficiently, keeps costs down and ensures that the resultant project accurately reflects the customers needs.

There are five phases to a Special Project, each has its own unique characteristics, and each constitutes a major milestone in the Project life. The five phases are:

Phase 1 - Preliminary Investigation.

The purpose of a Preliminary Investigation is essentially to determine whether there is scope for a project. This is not only from a technical viewpoint but also in terms of the potential costs relative to the business needs of the customer. The deliverables of

Phase 1 are a Functional Specification Proposal. Which includes a *ball-park* estimate of project scope and cost.

Phase 2 - Functional Specification.

This is the most important phase of a special project. It is in this that we conduct a detailed analysis of the project in partnership with the customer.

Typically this would involve one or more working sessions with the customer to determine needs and requirements and gain an in-depth understanding of the problem. This is combined with a period of technical investigation and specification at OPD.

If the results of the investigation reveal that there is a need for a special development, the deliverables of this phase are:

- Scope
  A detailed Functional Specification outlining the architecture, functionality and constraints of the solution and its operation within the proposed environments.

- Cost
  A fixed cost for the development of the project.

- Schedule
  An estimated development and delivery schedule.

These deliverables are encapsulated in a Software Development Project Agreement which sets out, for both parties, the terms and conditions of project development and documents the mutual understandings. In addition, the Specification phase produces:

- Support Plan

  It is an objective of all specials that, at the earliest opportunity, as much of the service as is appropriate will be incorporated into the standard products. This provides the customer with the same long term commitment to support for a special that they would expect for any standard Hewlett-Packard product. There may be components for which this is not appropriate or feasible. These components will be identified in the Functional Specification and may be either handed over, in source code, to the customer or supported long term by a Custom Support Plan (CSP). A CSP is normally put in place for the period after completion of the special up to the time it becomes part of the standard product so that immediately delivery takes place, the Customer has access to the full support resources of HP.

- Acceptance Plan

Since the customer is a major participant in the specification of the project, they must also be a major participant in the verification that the delivered product meets the specification. The last act in the development phase is acceptance testing and it is important that it is planned and understood as early as possible.

Towards the end of the preparation of a Functional Specification, HP and the customer review the project together. This review is an important process for not only does it give both parties the opportunity to discuss and evaluate the proposal but it also gives the customer (or HP) the opportunity to vary any of Scope, Cost and Schedule recognizing that each has an impact on the other.

Phase 3 - Development.
During the development phase HP works with the customer to develop and install the special. It includes all activity from Functional Specification approval to completed acceptance testing. It includes all those aspects of a software development life cycle that are included in standard product development. Test planning, internal design, development, testing, quality analysis progress tracking and reporting etc. are all as much a part of the specials development cycle as for any project. Although Specials tend in general to be smaller than a normal HP development project, they have no less need of a thorough and complete development process and attention to quality. Phase 3 concludes with acceptance tests and delivery.

Phase 4 - Warranty and Special Support.
Upon completion of the development and demonstration to the customer that the product substantially conforms to the Functional Specification, a warranty and Special Support phase commences. During the warranty period (which is typically 90 days), HP and the customer work together to rectify defects and to bring the product further into compliance with the Functional Specification. Special Support, for a period after warranty is optional and is typically covered by a Custom Support Plan (CSP).

Phase 5 - 'Roll-In'.
At the earliest practical opportunity, the Special Product is included in the HP Product Line. This provides the customer with the benefits of long term support and the use of the standard product enhancement process. There are often some components of the special (occasionally all of it) which cannot be included in a standard product. These components will have been isolated at specification phase and a plan for long-term support put in place. Normally the majority of the special will become part of a standard product so that the CSP can be terminated and the ongoing support costs to the customer reduced.

A Special Project Team

As with all projects, a special project is a team effort. However, Special project teams are of significantly different composition than a product development team. There are in fact two project teams, one from HP and one from the customers organization.

The HP project team that is brought to the customers facility is staffed by a Project Manager (PM), Office Automation specialist(s) from OPD and a local Office Automation software support engineer. This team is supported by the product development teams, the division and response center support organizations and the local sales and support team. The HP Project Manager assigned to the project is responsible for the activities leading to the production and delivery of the project deliverables. The HP Project Manager is the focal point of contact between Hewlett-Packard and the Customers organization for the project.

For the Specials process to work successfully, the appropriate people in the customers organization must also participate. As a minimum, a review body consisting of representatives who know the day-to-day operations, procedures and methods both in the MIS staff and the local and remote user community is required. These resources know what is necessary to make their individual responsibilities / functional areas more efficient and effective. It is very important to the project that the customers management identify, make available and communicate their support to the key participants in the review process.

As is the case with all projects, there are many activities which require constant attention. Hewlett-Packard provides a Project Manager (PM) to the Project. The HP Project Manager will drive and direct most of the activities relating to the project, but will need a single point of contact within the Customers organization. Therefore, prior to the start of any project activities, it is be requisite that Customers management select and empower a project manager (PM) to work peer-to-peer with the HP Project Manager. While the customers PM might not be a full time job, he or she must be available when needed to help schedule, coordinate, meet, report, communicate etc. the customers resources and activities.

Summary

The availability of a Specials capability within the Office Productivity Division has opened a number of opportunities for HPs customers. With such a capability the ability for a customer to address a business need with HPs Office Products is not restricted by the closeness of fit between the standard product and the customers needs. It is restricted, as it should be, by the ability of the customer to analyze and justify the return on investment. If a facility is required by a specific customer it only needs be justified in terms of their own organizational costs and benefits not those of entire installed base for the vendors products.

The Specials Life Cycle and process described above reflects the commitment we have to doing the job right the first time. It is our experience that to do it right both the customer and HP will need a basis for agreement as to the user's requirements and business goals for the project. HP believes that a detailed, agreed functional specification document is essential to the success of the project. The finished document provides the customers management with a level of detail necessary to ensure that project expectations are identified and possible to be achieved. It also serves as a project guide book and helps both parties readily identify issues in a timely and appropriate manner.

Cooperative Processing --
Making the Most Out Of the
PC - HP 3000 Connection

Bruce N. Smith
Hewlett-Packard Company
8000 Foothills Blvd.
Roseville, CA 95678

In the beginning...

In the beginning, the PC and the HP 3000 were separated by an impenetrable wall. When the personal computer was first used in the commercial environment, connectivity with departmental computers was non-existent. Any interaction between the two was typified by the accountant who copied needed numbers from a printed report into an electronic spreadsheet, then massaged them a bit, created some elementary graphics, and, if necessary, created transactions that were then entered manually into the HP 3000. File transfer between the PC and the departmental computer was virtually unheard of.

And then there were terminal emulators...

Finally, the wall started to crumble with the introduction of terminal emulators. The PC user was now able to use his PC for real work; as a terminal hooked to the host computer. Some users were knowledgeable enough about the databases that they used to select data with Query, write a report to an MPE file (without carriage control), copy the file to the PC using their favorite emulator software (AdvanceLink of course), and import the file into their favorite spreadsheet or database package. Again, after working with the data, they still had to formulate transactions for manual input on the HP 3000.

Today...

With the advent of "cooperative processing", the wall has now been torn down. Hewlett-Packard's Cooperative Services product (HPCS) gives the third generation language application designer the ability to create PC programs that:

- Directly read, write, and update HP 3000 TurboIMAGE data bases.
- Directly read, write, and update MPE files.
- Execute MPE commands
- Exchange data with host procedures stored in segmented libraries (SLs).
- Use local PC files, printers, and plotters.
- Access the host HP 3000 via OfficeShare LAN or basic serial connections.
- Use shared discs on HP 3000 and PC servers.

- Present a consistent PC user interface using MS Windows and NewWave or other company standard screen handlers.

- Allow user mobility by running from portable computers in the field or at home.

- Can be written in COBOL, Pascal, or C languages and run on the HP Vectra PC, the HP Touchscreen, the Portable Plus, or on the IBM PC/XT, PC/AT or PS/2 personal computers attached to almost any HP 3000.

What is Cooperative Services?

Cooperative Services is the only PC and HP 3000 software required to create PC applications written in COBOL, C, or Pascal that exchange data with the HP 3000 host.

The major components of the product are:

- The HP 3000 server process.

- The "Requestor" -- a terminate-and-stay-resident (TSR) PC process.

- A set of library modules that are linked with the application program.

- Conversion modules to handle the differences between character sets and numeric representations on the PC and the host.

Why would I want to write a cooperative processing application rather than a HP 3000 terminal-based application?

Cooperative processing applications are your best choice:

- If you have already written your PC application and you would like to have your program retrieve some data from the HP 3000 or write data to a host database, file or, possibly, send data to another user via HP Desk Manager.

- If your intended user base is already using PCs for word processing, graphics or with local databases and they would prefer not to use a terminal emulator program.

- If your users are geographically separated from the

HP 3000 and you want to reduce the costs of data communications by minimizing the connection time to the host.

■ If you want to automate data transfers from field personnel who use portable computers to collect and validate data on the PC prior to transfer.

■ To provide your PC community with a consistent user interface with tp data processing tasks. The most obvious case would be those who are used to MS Windows and, in the near future, HP's NewWave product.

Will Cooperative Services increase the performance of our HP 3000s?

In a well planned transaction processing application, where the data is stored both on the host and on the PC and access to the HP 3000 is kept at a minimum, cooperative processing will not degrade your HP 3000 performance. In some cases, performance may actually improve by removing the screen handling and data validation from the host machine.

Moving an existing application from the HP 3000 to the PC without changing where the bulk of the processing takes place, can actually increase the load on the host.

Several actions can be taken to reduce the load on the host:

■ Do as much processing on the PC as possible. This may require keeping certain fairly static data bases on the PC for validation instead of going to the host to check each item on a data entry screen. These validation files may be refreshed on a daily, weekly or monthly basis, depending on the needs of your application. The files would contain data such as account and department numbers, part numbers, or customer information.

■ Use Remote Procedure Calls (RPC) to do the heavier host processing. For instance, if a transaction requires reading through long chains and returning only a subset of all that was read, most of the work should be done in a procedure stored in the group or account SL. This way, the PC will only have to make one call to the host for information, instead of performing a DBFIND and many DBGETS. The requested data can then be returned in a single block.

■ An asychronous process can be spawned on the HP 3000 with an RPC to handle transactions as time allows.

Many PCs could write complete transactions to a message file, and the host process would execute them on a first come first basis. Each PC would continue immediately, as if the transaction had been posted. This would also give the PC user the impression that he had the entire machine to himself.

Is HPCS hard to use?

Writing PC applications to access host databases is not a lot different than doing it on the HP 3000. Here is an example of opening a host database from PC COBOL:

```
MOVE "  HCSAMP.PPCDICT.HPOFFICE;" TO BASE.
MOVE "MGR;" TO DB-PASSWORD.
MOVE 5 TO DB-MODE.

CALL "DBOPEN" USING BASE DB-PASSWORD DB-MODE DB-STATUS
                    HCSTATUS.

IF HCSTATUS NOT = 0
MOVE "DBOPEN" TO HC-CALL
PERFORM HC-ERR THRU HC-ERR-EXIT.

IF COND-CODE NOT = 0
```

The only difference between this and host COBOL is the addition of the HCSTATUS variable. HCSTATUS was included to provide the application the status of the data communications between the two machines.

Since data and not just addresses are sent to the host during IMAGE and MPE file access, the first word of each buffer must contain the size of the buffer in bytes. For example, in COBOL this would be the way to define a structure for a DBGET:

```
WORKING-STORAGE SECTION.
01  ORDER-MASTER.
    05  LNGTH            PIC S9(4)
                         USAGE COMP-0.
    05  ORDER-NUMBER     PIC X(10).
    05  CUST-NUMBER      PIC X(08).
    05  ORDER-DATE       PIC X(06).
    . . .
PROCEDURE DIVISION.
    . . .
    MOVE 24 TO LNGTH OF ORDER-MASTER.
    MOVE 2  TO DB-MODE.
    MOVE "*;" TO DB-LIST.
    MOVE "ORDER-MASTER;" TO DSET.
```

```
CALL "DBGET" USING BASE DSET DB-MODE DB-STATUS
                   DB-LIST ORDER-DETAIL DUMMY HCSTATUS.

IF HCSTATUS NOT = 0
. . .
```

It is also necessary to add code to make and terminate the connection with the host HP 3000. This entails providing the logon string and the server name for OfficeShare, or COM1 or COM2 for a basic serial connection. In COBOL using OfficeShare to connect to a HP 3000 named "SERVER" might look like this:

```
MOVE "SERVER" TO CONNECTID.
MOVE "MGR.SAMPLE" TO LOGON.

CALL "CONNECT" USING CONNECTID LOGON HCSTATUS.
IF HCSTATUS NOT = 0
. . .

MOVE 0 TO TRACING.

CALL "STARTSERVER" USING TRACING HCSTATUS.
IF HCSTATUS NOT = 0
. . .
```

Disconnecting is just as easy:

```
MOVE 0 TO DUMMY.
CALL "STOPSERVER" USING DUMMY HCSTATUS.

MOVE 1 TO LOGOFF.
CALL "DISCONNECT" USING LOGOFF HCSTATUS.
```

Data Format Conversion

The remaining difference between HP 3000 and PC code is converting data between PC and HP 3000 formats.

Numeric Format

Numbers were never created equal. For example, when the personal computer was created, all integers were stored in 8-bit bytes, the low byte on the left then the high byte on the right. Even with today's 16 and 32-bit architectures, integers are still stored in the same manner. The HP 3000, on the other hand, writes data in 16-bit words -- high byte on the left and the low byte on the right, just the opposite of the PC.

Because of these types of inconsistencies the CONVERT function has been provided to convert to and from the following PC and host numeric formats:

Pascal and C Conversion Pairs

Host Format	PC Format
TurboIMAGE I, J	8086 1 Word Integer
TurboIMAGE I2, J2	8086 2 Word Integer
TurboIMAGE R2	8086 2 Word Floating Point
TurboIMAGE R4	8086 4 Word Floating Point
TurboIMAGE P	8086 2 Word Integer
TurboIMAGE K1	Unsigned Integer

Microsoft COBOL Conversion Pairs

TurboIMAGE P	Packed Decimal
TurboIMAGE Z	Zoned Decimal
TurboIMAGE K1	COMP-4

Micro Focus VS COBOL Conversion Pairs

TurboIMAGE P	Packed Decimal
TurboIMAGE Z	Zoned Decimal
TurboIMAGE K1	COMP (2 word integer)

Numeric conversion is necessary only when numbers are brought down to the PC from the HP 3000 or are being sent to the HP 3000 in a buffer. HPCS has no knowledge of the format of data in the program's data areas. HPCS does automatically convert some HP 3000 data, such as certain words of the TurboIMAGE status array and error numbers returned by FCHECK.

As an example, all of the TurboIMAGE intrinsics return information in the STATUS array. When the intrinsic call is successful, Cooperative Services automatically converts selected words from host to PC format. Typically, record addresses are not converted, because their value is of little use to the PC application.

If TurboIMAGE call is unsuccessful, only the condition code is converted to PC format, while the other nine words are left in host format for use by the DBERROR and DBEXPLAIN intrinsics.

Character Formats

Data on a host may be represented in either ISO-7 or Roman8 formats. Data on a PC may be represented in Roman8, IBM8 or ANSI formats.

It is difficult to make absolute statements about patterns of character code set usage. However, the Vectra family and IBM PCs are more likely to use the IBM8 character set. The HP Touchscreen typically to uses the Roman8 character set. Microsoft Windows uses the ANSI character set. Individual software packages exhibit variability in code set use.

If you expect your application to have international distribution, you may want to consider providing the following types of character set conversions:

Asian Character Set Conversion Pair

Host Character Set	PC Character Set
Host Chinese or Taiwanese	PC Chinese or Taiwanese

ISO-7 Character Sets Conversion Pairs

ISO-7 Host Format	PC Format
US Swedish/Finnish Danish/Norwegian French German UK Spanish Italian	Roman8 IBM8 ANSI
PC or Host Format	**Host Only Format**
Roman8 ANSI	IBM8
IBM8	ANSI

Conclusion

Cooperative processing is not for all possible applications. But if you are looking for a way to add value to your new or existing PC applications, to present a consistent PC user interface to your users and customers, or give mobility to a force of sales or service personnel, now is the time to consider creating cooperative processing systems.

MATCHING PRINTER TECHNOLOGIES TO YOUR OFFICE NEEDS

LARRY TRACY
HEWLETT-PACKARD
MARKET DEVELOPMENT
BOISE DIVISION/OFFICE PRINTERS
11311 Chinden Boulevard
Boise, Idaho 83714

The purpose of this paper is to enlighten the reader on various printer technologies as well as provide some insight to where each of these technologies can be best utilized.

Before this can be effectively accomplished, it is first necessary to discuss four common business printing areas. These areas are:

A) DeskTop Publishing
B) Business Graphics
C) Office Publishing
D) Electronic Data Publishing

A) DESKTOP PUBLISHING

Desktop publishing applications range from the simplest documents (text only) created with electronic typewriters or word processors to documents consisting of any combination of test, graphics, charts, illustrations, photographs and numeric data. Desktop publishers are now producing camera-ready artwork for manuals, newsletters and other formal communication using an integrated page composition system without ever making a trip to the typesetter. During the past 20 years, computers have automated text creation in the office from labor intensive manual methods straight into the twentieth century. Computer systems designed for processing text save time, effort and money: more critically, they give businesses more and more control over all aspects of the appearance and production of their text documents.

Categories Within DeskTop Publishing Include:

Professional Typesetting: - Applications include camera-ready documents, and final drafts for publishing.

Page Composition: - Applications include manuals, press releases, newsletters, price lists and user documentation.

Merged Text & Graphics: - Applications include proposals, technical documents, text printouts from word processing, spreadsheets and database software integrated with graphics and presentation materials.

Sophisticated Word Processing: - Output typically includes contracts, articles, short manuals and external correspondence.

General Word Processing: - Output includes letters, interoffice correspondence, spreadsheets & reports.

B) BUSINESS GRAPHICS

Every day, American offices generate more than 600 million pages of computer printouts. A survey published by Dewar's Career Profile showed that 42% of respondents cited excessive paperwork as the major data processing problem. The information systems of today are capable of generating reams of "timely" data designed to assist with decision making. Business graphics offers a viable alternative to the problem of information overload.

Studies suggest that a person can absorb tables of numbers at 600 to 1200 words per minute. By comparison, a person familiar with reading pictures, charts and graphs can comprehend information at a rate equivalent to 50 to 70 million words per minute. These and other claims for increased efficiency and productivity are verified by the rapid growth occurring in this market.

Business Graphics Applications

Applications in Business Graphics include:

Decision Graphics: - The use of charts for analysis and decision making. Analysis of spreadsheet data, past and present, for example, is used to perform "what if" transformations.

Information Graphics: - The use of graphics for interdepartmental communications and reports. Its primary use is communicating a point more efficiently, rather than serving as an analysis tool.

Presentation Graphics: - The use of graphics to provide visual output during presentations such as overhead transparencies, flip charts or 35mm slides.

C) OFFICE PUBLISHING

Corporate publishing systems are generally found in larger corporations (Fortune 1000 companies) that produce high-quality documents at high volume as a regular part of their business. Typical publications include reports, manuals, annual reports, prospectuses and newsletters. Intended for prospects, clients, and shareholders, these publications must convey a very high quality image. To save money and maintain better control over these publications, many companies often choose to bring publishing systems in house. Such systems also offer corporations many of the same publishing tools which are expensive when using outside typesetting and graphics professionals.

Corporate publishing solutions available on multiuser computer systems integrate mini-computer or mainframe hardware with text, graphics and page composition software. It couples the speed and processing power of a large computer with the ability to link users to a corporate database (as well as provide information from one user to another). High-end laser printers linked to multiuser systems allow for high-speed printing of compound documents. The higher price tags of these laser printers can be offset by increased productivity for a larger number of people.

D) ELECTRONIC DATA PUBLISHING

EDP printers are usually connected directly to mini or mainframe computer systems in an Information System Department. These printers are used for high volume printing jobs and internal correspondence, since most EDP printers have resolutions too low to produce high quality output necessary for for office publishing.

Typical applications for EDP printers include operator technical manuals, accounting output such as large spreadsheets, general ledgers & balance sheets, rough drafts, system dumps, electronic mail messages, printing of pre printed forms, barcoding for shipping and receiving docks and printing of ordinary documents that may need to be archived.

Although EDP printers may have a higher initial cost than office system printers, they typically have a lower cost per page. For example, a 45 page per minute laser printer has a cost of about 1 cent per page, but an 8 page per minute laser printer will have a cost of about 3 cents per page (cost per page based on consumables only, paper was not included).

Duty cycle is another area where EDP printers and office printers differ. EDP printers typically have higher duty cycles; up to about 1 MILLION pages per month. Office system printers have duty cycles of less than 100,000 pages per month. This is why EDP printers are often the computer system's "workhorse", used for large volume printing and the day to day rough drafts.

Now that we have examined the different applications for printers, let's discuss the technologies which dominate the market place today. Raster printing devices like serial dot matrix, line impact dot matrix, inkjet and laser printers provide for most of the office printing needs.

RASTER PRINTERS

The achievement of the mid-seventies toward the advancement of computer graphics was cheap raster graphics based on television technology. In raster displays, the display "primitives" such as lines, characters and solid areas are stored in a refresh buffer in terms of their component points, called pixels or pels (picture elements). The image on the screen is formed from the raster, a set of horizontal raster lines made up of individual pixels.

The concept of raster also applies to raster printing devices. The raster is simply a matrix of pixels covering the entire area, whether a screen or piece of paper. The entire image is scanned sequentially, one raster line at a time, top to bottom.

The storage needed is greatly increased because each pixel must be stored in a refresh buffer as a "bit map" containing only points that map one for one to points on the screen. The development that made raster graphics possible was solid-state memory which provides refresh buffers considerably larger than those of a decade ago at a fraction of the price. All of the pixels in a primitive such as a line or rectangle must be transformed in the buffer to their new coordinates, rather than just the end points of lines, as in vector plotting. Because of the heavy memory demands of graphics applications, much more intelligence is being downloaded into the hardcopy device to relieve much of the computing burden. A number of graphics-oriented printers are more powerful computers than their hosts.

How does all this relate to printing? The first requirement enabling a printer to print bit mapped graphics is a dot matrix form of printing as opposed to fully formed character printing. Normally, the host computer sends the printer a code for a character. The printer has a ROM memory chip called a character generator and the program in this memory establishes the pattern for every character in the set. A printer with provisions for bit-mapped graphics generally recognizes a certain code sent by the computer as an instruction to turn off the character generator and bypass the print logic that controls the printing of individual dots. The printer then interprets the data stream following the turn-off code as explicit orders to print certain dots. This bit mapped control permits the printing of a pattern of dots on the paper to form a picture or graphic image.

Raster printing devices may be categorized by printing technologies, impact or non-impact. Some examples in each category are:

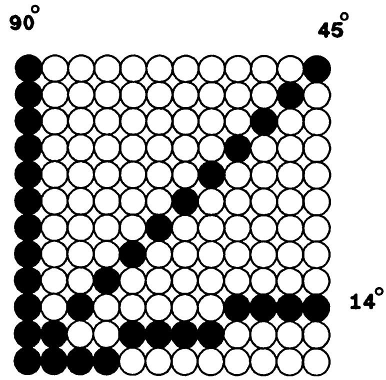
Impact:
* Serial Impact Dot Matrix
* Line Impact Dot Matrix

Non-Impact:
* Inkjet
* Thermal
* Laser
* Ion Deposition

PRINT QUALITY

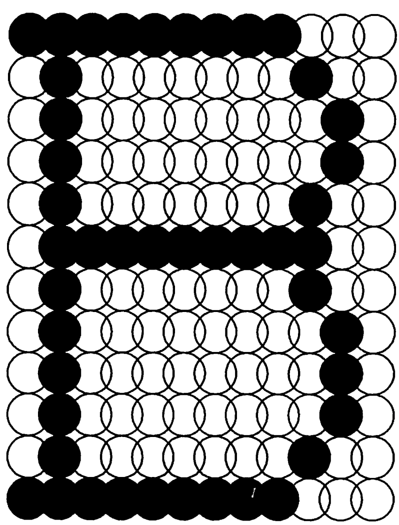
The output from a raster device is an array of dots and the resolution is determined by the number of dots-per-inch (dpi) or points per inch (ppi). These hardcopy devices are usually refereed to as "dot matrix". The term "matrix" refers to this raster pattern and the term "dot" obviously refers to the spots of ink or toner that form this pattern.

The figure below shows a magnified view of raster output. The dots can be darkened or left blank. For comparison, three lines are represented: one at 90 degrees, another at 45 and one at an angle close to the horizontal axis, 14 degrees. Notice how the quality of the line depends on the angle at which it is drawn. The closer to vertical or horizontal, the more obvious the "scalloped" effect becomes.

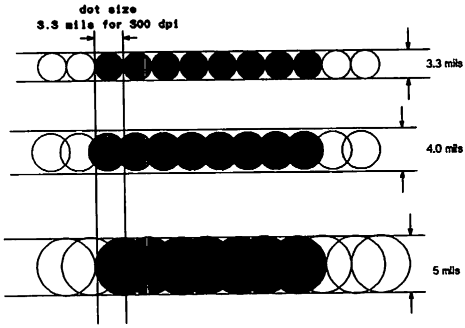


To achieve print quality using a raster device, the scalloped effect must be minimized. This can be accomplished using various techniques. The most straightforward technique is to control the distance between dot centers and thereby controlling the resolution (dpi). Raster hardcopy output devices vary in resolution from under 80 dpi to greater than 400 dpi.

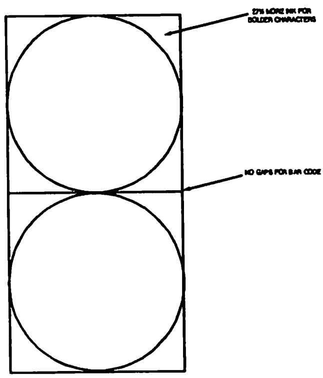
The cell in the following illustration in seven dots wide and nine dots long. Notice that the dots in each row overlap. This is known as "half dot shift". Shifting dots slightly allows for rounder curves and more readable letters.



The second technique for minimizing the scallop is to control the dot size. The dot size determines line width and, depending on dot spacing, how much overlap will occur. Increasing dot size without changing resolution gives more overlap and smoother edges, but also creates a wider line, as illustrated below.



A third technique for minimizing this scallop effect is to alter the shape of the dots from round to square, as determined by the shape of the print stylus. Square dots minimize the scallop effect, providing more readable characters, and better line drawing and bar codes



The combination of resolution, dot size and dot shape are elements in determining the print quality.

THROUGHPUT

Different raster output technologies specify the speed of a raster device in various measurements.

Devices which create characters with a moving head measure speed in characters per second (cps). Those which print one dot row at a time to form characters and graphics specify speed in lines per minute (lpm). Devices which format and print entire pages of text and graphics at one time specify speed in terms of pages per minute (ppm). Devices used primarily for graphics specify speeds in inches per second (ips).

RASTER PRINTING TECHNOLOGIES

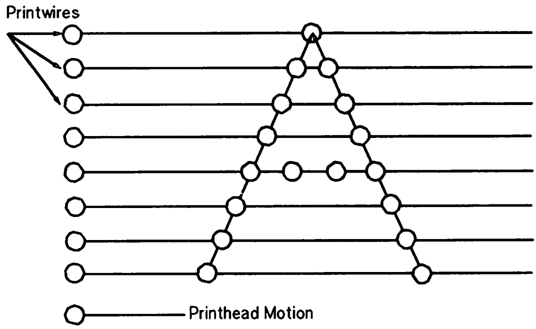
Two types of raster printing technologies will be outlined. Impact printing uses a mechanism that touches the paper and leaves ink on it. Non-impact uses a mechanism that either exposes, charges or sprays the print on the paper without actually striking the paper.

IMPACT

Serial Impact Dot Matrix

The basic method for forming characters with a serial impact dot matrix printer is to move a vertical column of print wires across a line and to strike the paper through an inked ribbon. Each time a wire hits the ribbon, it leaves a dot on the paper. Each wire on the printhead can be driven at over 1000 times per second to form a character within a matrix cell. The printhead uses tungston rids attached at one end to small solenoids and springs. The other end of the wire passes through a wire guide where the wires meet the ribbon and paper. Print quality and formation depend upon the number of wires (usually between 9 and 24), wire speed and the internal control logic of the printer. Speeds on this type of printer ranges from 45 to 360 characters per second and typically the higher the speed the lower the character resolution will become.

Serial impact dot matrix printers are the most common printer in today's office. These printers are normally found printing internal memos, spreadsheets, accounting reports, payroll checks, low resolution graphics, etc. Due to the low resolution of these types of devices, documents generated with serial impact dot matrix printers are rarely used for customer letters, presentation graphics or other uses where a truly polished appearance is necessary.



Advantages:

- Multiple character sets
- Graphics
- Multipart forms
- Low cost per page
- Low initial purchase price

Disadvantages:

- Lower speeds
- Noise
- Lower resolution

Average Cost:

$300 - 1500

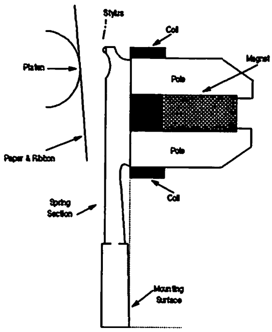
Monthly Print Volume:

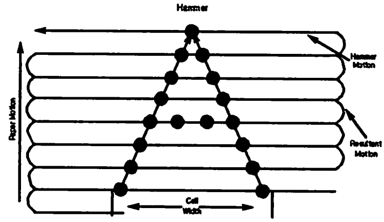
100 - 3000 pages

Line Impact Dot Matrix

In line impact dot matrix printers, the print tines or hammers are mounted on a printbar which moves horizontally. The print bar vibrates from side to side to allow one hammer to print from 4 to 16 dots in each horizontal row. A small dot or stylus is mounted on each hammer. The hammer is held back by magnetic force, and when neutralized, is thrown forward by the force of a spring. The hammer impacts the ribbon and paper and is then drawn back to the "loaded" position by magnetic force. Each character is formed one dot row at a time as the paper advances in a smooth motion.

Line impact dot matrix printers range in speed from 300 lines per minute up to 1600 lines per minute. This type of printer is the standard workhorse printer in most datacenters, and more recently, this type of printer can be found as a shared printer on a local area network or as a remote printer in a small department. Printout ranges from EDP reports to barcode labels for inventory control to multipart checks.





Advantages:

- Higher speeds
- Graphics
- Multipart forms
- Low cost per page
- High print volumes

Disadvantages:

- Lower resolution
- Noise
- Pin Feed Paper only

Average Cost:

$4,000 - 26,000

Monthly Print Volume:

3000 - 30,000 pages

NON-IMPACT

Inkjet

Inkjet printing is a broad term describing a form of printing in which drops of ink are projected onto a surface using a variety of techniques. There are two distinct inkjet technologies today, they are: Continuous inkjet and Thermal (sometimes called Drop On Demand) inkjet.

Continuous Inkjet

Continuous inkjet printers will produce a steady stream of magnetically charged ink drops. These drops are passed thru a magnetic field which will either guide the drop on to the paper or into a recycle catch tray. The recycled drops are then passed thru a filter to remove any stray paper dust or other contaminants before they are pumped back thru the inkjet head. Because ink is continuously pumped thru the print head, the small ink nozzles will clog less frequently.

This type of printer is not often found in the office environment. Due to the somewhat high cost per page of this device, this application is better suited for applications where a high volume of paper, but low volume of characters (ex. address printing on envelopes) is required.

Advantages:	Disadvantages:
High resolution depending on drop size	Pumps can be noisy
Flexible character design	May require special papers
Graphics	No multipart forms

Average Cost:	Monthly Print Volume:
$1,500 - 25,000	Not Available

Thermal Inkjet

In the Thermal Drop on Demand Inkjet printers, ink is held in a small reservoir that is an integral part of the printhead. Capillary action forces ink into tiny channels behind the printhead. When an ink dot is required, an electric current heats up a thermal resistor. The thermal resistor "boils" a drop of ink which squirts onto the paper. Because ink is sprayed only as needed, there is no need for filters or bulky ink pumps. The typical speed for TIJ (Thermal InkJet) printers is between 120 characters per second and 2 pages per minute.

These printers are now capable of producing 300 dot per inch characters and graphics (near laser printer quality). Thermal Inkjet printers are typically found used by a single user in an office where quite printing is a necessity. Typical printout includes: spreadsheets, interoffice correspondence, business graphics, screen dumps and other low volume office printing. These printers usually produce less noise than a normal office conversation (less than 50 dba).

By combining a black printhead with a printhead containing the three primary colors cyan, magenta and yellow, a user can now produce color documents. Today's P/C software packages are just now beginning to merge color in with black and white printing to produce spreadsheets with negative numbers in red. Other packages will allow the company logo to be printed in color, the body of text to be printed in black and the pie chart to be printed in color.

Advantages:	Disadvantages:
High resolution depending on drop size	Limited speed
Flexible character design	May require special papers
Graphics	No multipart forms
Color Printing	
Quiet operation	

Average Cost:	Monthly Print Volume
$ 500 - 2,500	100 - 3000 pages

Laser

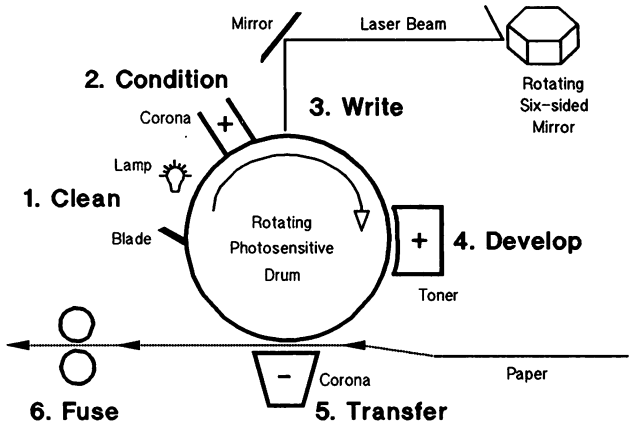
Laser printers are part of a family of non-impact printers which are "electro-photographic". Electrophotography refers to a copying or imaging process in which toner is attracted to portions of a photosensitive plate, drum or other intermediary.

Laser printers use a mechanically deflected beam that has been modulated (turned on or off) with print data to trace the "page" as an electrical image on the drum. In reality, one dot at a time is written as the beam is swept across the face of the photosensitive drum via the polygon mirror. Toner is then attracted to the charged areas of the drum (the areas which were charged by the laser). The toner is then "transferred" to the paper by a large electrostatic charge. The loose toner and paper are then passed thru a fuser which melts the toner into the paper.

Laser printers range in speed from 5 pages per minute to 200 pages per minute. Laser printers are broken up into three different classes: Desktop (5 - 12 pages per minute), Departmental (15 - 30 pages per minute) and EDP / Datacenter (30+ pages per minute). Resolution on laser printers ranges from 180 dots per inch to 600 dots per inch, with 300 dots per inch found in the most common laser printers.

Laser printers perform a wide variety of office printing needs. These printers can be found printing spreadsheets, letter (laser) quality correspondence, business graphics, CAD (Computer Aided Design) graphics, DeskTop Publishing as well as a multitude of other office printing. Due to a higher initial purchase price than most other printers, laser printers are typically shared by two or more users by use of an electronic switchbox (Note: Most laser printer manufacturers do not advocate the use of manual switchboxes as it can cause damage to the internal circuitry of a laser printer), local area network, minicomputers and mainframe computers. Some laser printers work only with cut sheet paper, others work only with continuous forms and a very small portion of laser printers have the ability to be converted from cut sheet to continuous form.

The Laser Process



Advantages:

- Excellent print quality
- Multiple character sets
- Electronic forms
- Graphics
- Quiet operation
- High speed

Disadvantages:

- No Multipart forms
- Higher initial purchase price

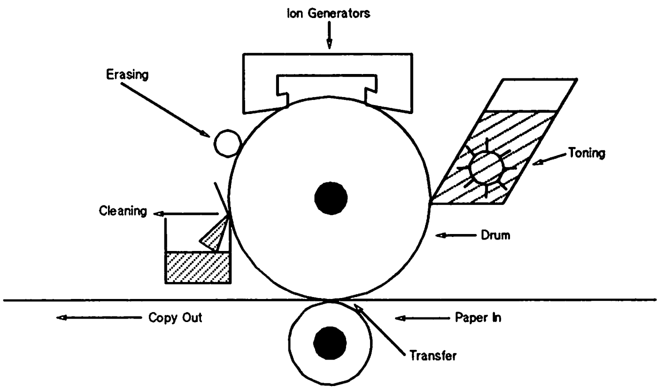
Average Cost: / Monthly Print Volume:

Desktop	\$ 2,000 - 10,000	4,000 - 12,000 pages
Departmental	\$15,000 - 30,000	10,000 - 100,000 pages
EDP	\$30,000 - 250,000	50,000 - 1,000,000 pages

Ion Deposition

Ion deposition devices are also electrophotographic and place a charge pattern corresponding to the desired image onto the dielectric surface of the drum. The "charging" action is accomplished through a non-contact ion projection cartridge which consist of a multi-plexed matrix of electrodes. The air contained in each cell is ionized when a voltage pulse is placed across the electrodes creating a pool of free ions. An electric field is then used to extract ions from the pool and accelerate them toward the drum. Negatively charged ions are attracted and positively charged ions are repelled from the surface of the drum. As the drum rotates, toner is attracted to the charged pattern on the drum. The image is transferred to the paper and fixed in place by pressing the paper between the image carrying drum and a lower pressure roller. The image is erased from the drum by slightly shaving the metal drum surface. Currently ALL ion deposition engines are manufactured by one company, Delphax.

Ion deposition printers are utilized in the same areas where departmental and EDP laser printers are found. Ion deposition printers are somewhat lower in cost to produce, but the print quality can be poor. Since the toner is pressure bonded, toner can sometimes be removed from the paper by rubbing or folding the paper.



Advantages:

- High speed
- Flexible character design
- Graphics
- Lower hardware costs
- Cut Sheet paper

Disadvantages:

- Poor durability of print
- "Shiny" print and "fat" characters due to pressure fusing
- No Continuous forms

How Electronic Forms are Changing Office Printing

Clay Young
Hewlett-Packard
11311 Chinden Blvd.
Boise, Id.

Table of Contents

1 Introduction

The efficiencies associated with electronic forms systems have made them so attractive that many organizations, large and small, are actively working on implementations. The following discussion is targeted toward assisting people in their understanding of the "world" of electronic forms. From historical aspects to modern day reality, the focus will be on forms and how to make this basic communication tool more efficient, productive, and cost effective. Also, the discussion will focus briefly of some of the elements that should be considered when evaluating an electronic forms system. A comprehensive look at the electronic forms world is beyond the scope of this document; however, a general understanding should be easily derived from the material presented.

So as to ensure that we all start with the same understanding of the elements of this discussion, the following key definitions are provided:

Form - the basic business tool for collecting and transmitting information, the catalyst for getting things done, and the record of what was done.

Electronic Forms - for the purpose of this discussion, electronic forms are images displayed on the computer screen. Where the the screen is filled in by an operator or the computer itself and then printed.

The use of electronic forms will be considered from the following perspectives:

- o Traditional Forms Development and Management;
- o Model for Electronic Forms;
- o Components of an Electronic Forms System; and
- o Determining What You Want to Accomplish.

2 Traditional Forms Development and Management

Historically, forms have been developed and managed via tedious, time consuming methods. To demonstrate this idea consider the following process, one representative of traditional forms development and management:

Concept

A need is identified or an idea born that can be most suitably catered to through the use of a form. A certain quantity of time is consumed at this stage determining if a form is actually the most productive way to handle the need or display the idea. If a form is deemed desirable for this application, comprehensive analysis must take place.

Analysis

The analysis stage is very critical because the form will be used as a basic business tool for collecting and transmitting information. To highlight the complexity of proper forms analysis, consider that by definition forms analysis is the systematic execution of those steps necessary to assure the following:

- o Productivity is increased in preparation, use, filing, and retrieval;
- o The total number of forms within the system is minimized;
- o Data element relationships are apparent through consistency and adherence to standards;

- o The effectiveness of the entire system , as well as the individual form, is enhanced; and
- o The resulting business tool communicates.

As is evident from this definition, forms analysis has been and still is a critical and extremely time consuming part of developing and managing forms.
Once the analysis stage has been successfully completed, the design of the form can begin.

Design

Traditionally, the design stage entails layout, proofreading, and approval. Of these activities, design is the largest time consumer - unless the organization must deal with an extensive approval process. Typically design has been shopped out or done in house with "crude" tools such as ruler and pencil and then taken to a typesetter. Once the design task is complete, production must commence.

Production

The production stage commonly includes the ordering, printing, and assembly of the desired form. Considerable time is required at this stage - often just waiting for forms to come back form a production facility. Once a production run comes back, inventory and storage become an issue.

Inventory/Storage

Traditionally, receiving, warehousing, requisitioning, and distribution are the major aspects of inventory/storage management. These are often the most costly elements of managing forms. Floor space is consumed and individuals need to be dedicated to managing the inventory aspects.

Processing

Processing consists of data entry, approval, filing, retrieval, transmittal, distribution, referencing, and copying. This is the stage where many people throughout the organization get involved with the form. The forms would be filled out and distributed manually. Once a form has been used, it often must be filed or maintained.

File/Maintenance

Some of the key elements of file maintenance are storage, retrieval, transfer, retention, and disposition. Historically, each of these have been done manually. Therefore, making the associated time and costs quite high.

In general, the traditional methodology of forms development and management have been time consuming, slow, and costly. A search for alternate solutions should lead a person directly to the concept - and reality - of electronic forms. The next section discusses the "new" way of dealing with the development and management of forms.

3 Model For Electronic Forms (adopted from the CG Corp. model)

As electronic forms have become a reality the forms development and management process has become significantly simplified. To demonstrate, consider this model for electronic forms in relation to the process previously described. Note: the elements of concept and analysis still exist in this model; however, since these areas have not been dramatically improved via the adoption of computers they are omitted.

Forms Design

In the world of electronic forms, forms design is computer aided layout and typography. Also, text, rules, and content are described at this stage. Further elements that make the form intelligent or linked to other forms can be added at this time.

Forms Integration/Conversion

During this stage, the necessary graphics elements are scanned and integrated into the form. Elements such as logos and signatures are common. Another scenario has a preprinted form being scanned and converted through a tracing function. Although this is time consuming, it is far less so than a total redesign. During this stage, completed forms are converted into files compatible with the main forms processing system and prepared for uploading. Also, a printer file is downloaded to the printer to await incoming data.

Communications

Communications links between the creation/integration workstation and the main forms processing system allow the form, or a reasonable facsimile thereof, to be uploaded for "mass" processing. This form is simply an interface for data entry, not a piece to be passed on to the printer with each print job.

Data Collection

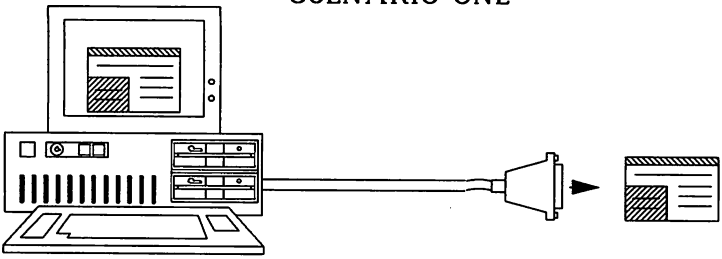
Terminals or integrated personal computers collect data from a data base or from screen entry and merge this variable data into the form waiting at the printer. Note that the data being sent around the system is ASCII data.

Output

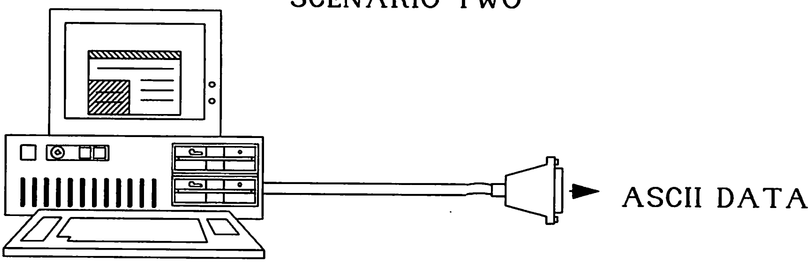
Completed forms are printed on demand by sending the ASCII data to the form waiting in the printers memory (downloaded early in the process).

Consider the following illustrations:

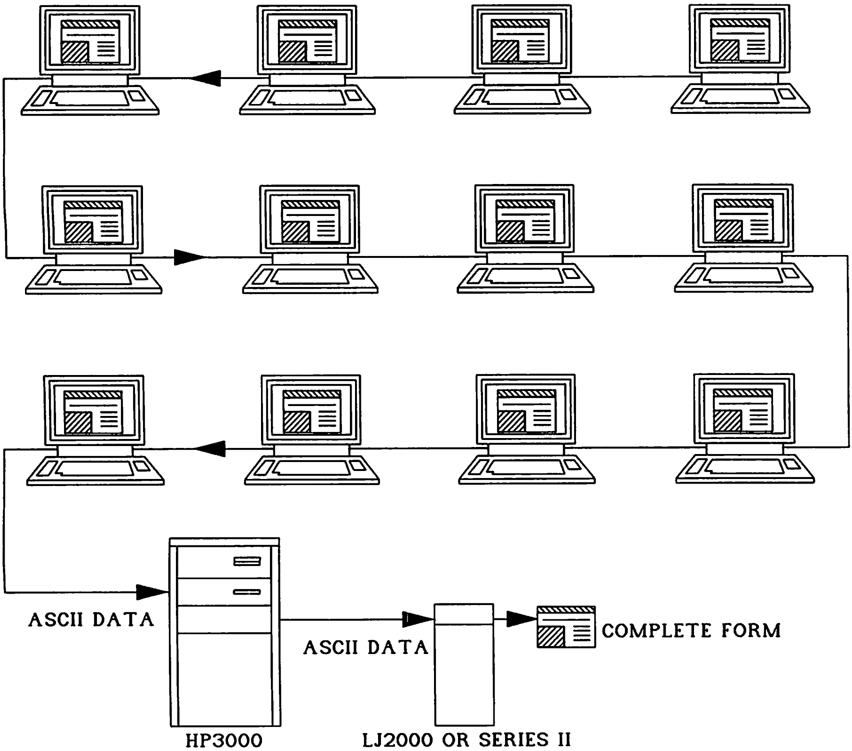
SCENARIO ONE



SCENARIO TWO



In this illustration, scenario one demonstrates a completed form being passed along to the printer. Scenario two assumes that the form has already been downloaded to the printer and that all that must be passed to the printer is the variable data. Of the two scenarios, number two is more efficient for jobs requiring more than one printing of the same form; primarily because the time to download just the ASCII data is less the time to download the form and the ASCII data for each job.



This illustration depicts an ideal configuration for a forms production environment. The user interface is a facsimile of the form, so the user can accurately place data in the appropriate fields. Once the form is completed and sent for printing, ONLY THE ASCII DATA will be passed down the line. The appropriate form (which was previously download to the printer) will pick up the data as it comes into the printer and a completed form will be printed.
Note that the same process could be occurring via data base input rather than user input.

As we can see, electronic forms generation differs quite dramatically from the traditional processes. What we should also be able to derive from this model is a tremendous number of efficiencies. Timeliness is dramatically improved, inventory costs are reduced, accuracy is improved, forms management is made easier, and so on.

4 Components of an Electronic Forms System

In order to obtain the greatest benefit from an electronic forms system, the proper system components must be in place. The next section describes, in greater detail, the elements of an electronic forms system.

An idea of what the major components of a forms system are can be acquired through the illustrations previously presented. However, for the sake of clarification, a description of the key components of an electronic forms system follows.

PC

A high speed workstation with advanced graphic capability is required for the forms design process. This workstation could be as sophisticated as a dedicated CAD station; however, that type of computing power should not be required. A typical forms design system should require no more than a 286 based PC with hard disc, graphics display, serial port, parallel port, and a mouse.

Mini/Mainframe System

The typical configuration would be multiple terminals hooked to the system for user entry or data base entry. The forms are designed on the PC workstation and passed on to the system for integration into the forms production environment. The fields associated with the forms are displayed to screen for easy, less error prone entry. Or, data can be acquired directly from the system data base; where no on screen entry is required. Optimally, the system contains the most up-to-date version of the form. Due to everyone obtaining the new form simultaneously, smooth and timely forms rollover is ensured. Consideration should be given to the idea that a forms system could as easily be a PC network. PC networks are quite common and can be used in a similar fashion as the mini/mainframe.

Software

The software used to design and manage forms is the most important aspect of the entire system. To ensure that software meets the basic requirements of forms design and management many aspects need to be considered. Further expansion of the subject is necessary to describe the more important elements of both forms design and forms management systems:

Forms Design:

The following are key attributes of an adequate forms design application:

- o High performance
- o Ease of use
- o WYSIWYG
- o Form conversion capabilities
- o Support for scanned images (logos, signatures)
- o Support a complete set of graphics elements
- o Support for LaserJet forms primitives
- o Support of bit-map graphics
- o Font variety
- o Font justification
- o Word wrap
- o Spell checking

- o Carbon black out
- o Build in security
- o Develop multi part forms
- o Inclusion of bar codes
- o Separate module

Forms Management:

The Following are key attributes of an adequate forms management system:

- o Ease of use
- o Data base merging capability
- o On screen data entry
- o Sophisticated filing functions
- o File portability
- o Support for multiple operating systems
- o Ability to merge variable data into a printer based, static, form
- o Forms security
- o File compatibility
- o Separate module

Each of these elements play an important role in a successful forms application. Many other aspects offer less significant value to the system; however, these aspects are beyond the scope of this general discussion.

Laser Printer

Laser printer technology has reached a point where the image quality is quite acceptable for most forms. There are many low cost laser printers available; however, very few are well adapted to the forms environment. Consider the key features of a printer that has been designed to offer advantages to the electronic forms world:

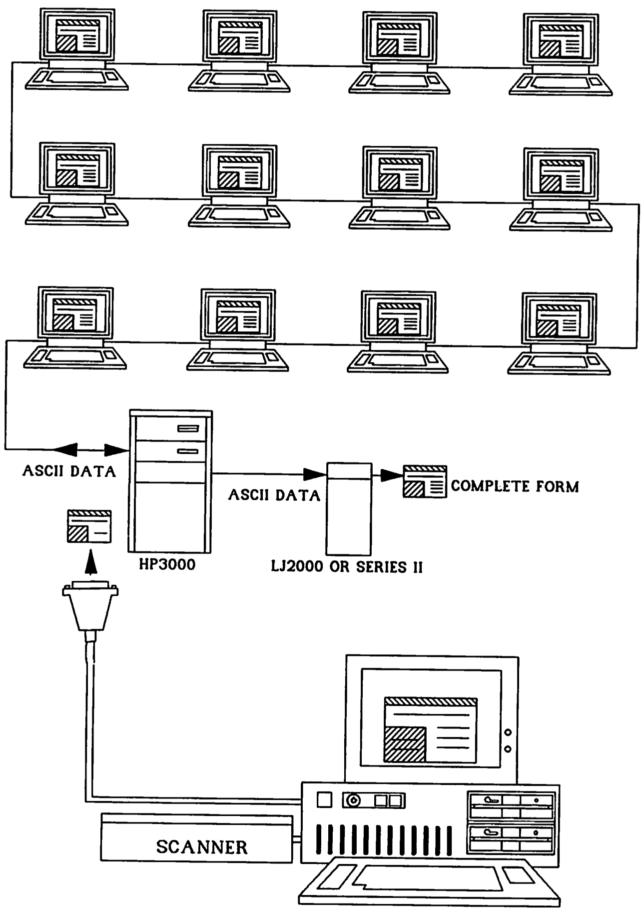
- o Built in forms design primitives such as lines, grey scale, and patterns
- o Macro handling capabilities for efficient processing of batch jobs
- o A full line of forms specific fonts
- o Easy to use
- o Broad range of software support
- o Compatible with other printers in a product line
- o Guaranteed file compatibility

The last item in the list is possibly the most important. If the printing device is constantly changing the way it handles features, the forms files that have been designed will require rework. A situation such as this is unacceptable because it places the user in the awkward mode of updating the form for each new device.

Scanner

The scanner is the fundamental tool used to bring in data and images that currently do not exist on the system. Scanners are used in the design aspect of electronic forms; where images are being transported into the form from an outside source. Since design is the focus of the scanner, it is typically connected to the design workstation. Note that scanners should not be used to transform a preprinted form into a computer form; simply because scanners produce very large binary files.

The following diagram illustrates a typical electronic forms system:



Additionally, a standard system would likely have a scanner for input and a personal laser printer for proofing the forms output - Prior to uploading it to the system or network.

5 Determining What You Want to Accomplish

While considering the implementation of an electronic forms system, determining objectives is key. The remaining discussion focuses on certain of the key aspects that may help in the evaluation of an electronic forms system.

Design Only

If the objective is to elevate the burden (slow layout, long lead times for revisions, and high production costs) placed on the organization by a manual forms design system, than a complete electronic forms system is not the most appropriate tool. However, since good forms system software is modular the design function can be purchased as a stand alone piece. This allows for forms design of the type described previously, while setting the stage for potential future growth into a comprehensive system.

Reduce Cost Associated With Traditional Paper Forms

Preprinted forms are very costly to maintain and are often considered likely candidates for computer automation. Often the objective of an electronic forms system will not be focused strictly on design, but rather how to reduce the costs associated with forms already being used. The cost that are traditionally associated with preprinted forms are as follows:

- o Processing time
- o Paper and printing costs
- o Obsolescence
- o Warehousing and out of stock

Each of these cost can be dramatically reduced through the power of a properly implemented electronic forms system. Consider how the associated costs can be reduced; long processing times associated with "wander-net" are reduced via the speed of electronic interchange; paper and printing costs are minimized since nearly all aspects are handled inhouse; forms that have become obsolete can be quickly updated and produced and minimal inventory of the old forms will be on hand; and, floor space and rush orders costs will be greatly reduced by having the system online. If the objective is to reduce such costs, a more comprehensive electronic forms system should be employed

Establish Demand Printing

If the the objective is ensure that information is distributed then printed, rather than the reverse, a fully functional electronic forms system should be considered. To establish demand printing, all the elements of the system should be in place. This is to ensure that the organization has control over the information flow, from the basic transmission device (the form) to the way the information is acquired (screen, data base) and printed. The benefits associated with computer design and reduction of costs are also present but are not considered to be the key attraction of the system.

React More Quickly to Business Needs

If the organizational environment is constantly changing, and in turn so are the forms being used, the implementation objective might be to react more quickly to that environment. An electronic forms system will help reduce the time devoted to printing, improve the organizations ability to comply with changing government regulations and tax laws, and help in responding to company restructuring, departmental changes, and acquisitions. The degree of electronic forms integration required to meet this objective will vary depending on the organizations environment.

Level of Implementation

To a certain extent the components that are chosen will depend on the desired level of implementation. Basically there three possible levels; personal, departmental, and organizational. For example:

- The personal level would likely be comprised of a PC, software, scanner, and a laser printer.
- The departmental level might consist of a network, software, a centralized design center (PC), scanner, and a network (distributed) laser printer.
- The organizational level may require a mini/mainframe, integrated software, a centralized design and management center (PC and system), scanner, and multiple laser printers - connected to both workstations and the mini/mainframe.

6 Conclusion

What has been presented here just scratches the surface of the electronic forms world. Hopefully this discussion has provided some insight on the history and current potential of forms design and management. While evaluating electronic forms systems, an organization should go far beyond this general discussion. Needs should be evaluated, system requirements defined, and appropriate forms system components investigated.

Resource Sharing: A Decentralized Processing Solution for Un-tapped Office Productivity

Tracy Crowe/Ann Pirrone
Hewlett-Packard Co.
Office Systems Division
8000 Foothills Blvd.
Roseville, CA 95678

I. OVERVIEW

Traditionally, one of the main factors affecting the productivity of an office workgroup is the availability of resources. The phrase 'availability of resources' can include anything from the number of people in the workgroup, to the number of letter-quality printers. For this paper, the term 'resources' refers to the parts that make up the office computing environment. Examples of these are: personal computers (PCs), host computers, terminals, datacommunications, disc drives, tape drives, software (program and data files), printers, plotters, and systems operations personnel. Any office having these resources has taken a major step to improve their productivity. This paper addresses these offices, and should bring to light additional steps that can be taken to further increase productivity through the sharing of resources.

II. THREE VARIETIES OF OFFICE COMPUTING ENVIRONMENTS

Resource sharing is not a new idea. It has been around for years. Many computer users have used it, although not always to their knowledge. Before discussing how a system's resources are shared, let's examine how they are connected to form the system. Here are three schemes for setting up office workgroups.

1. The first figure shows the most traditional workgroup setup.

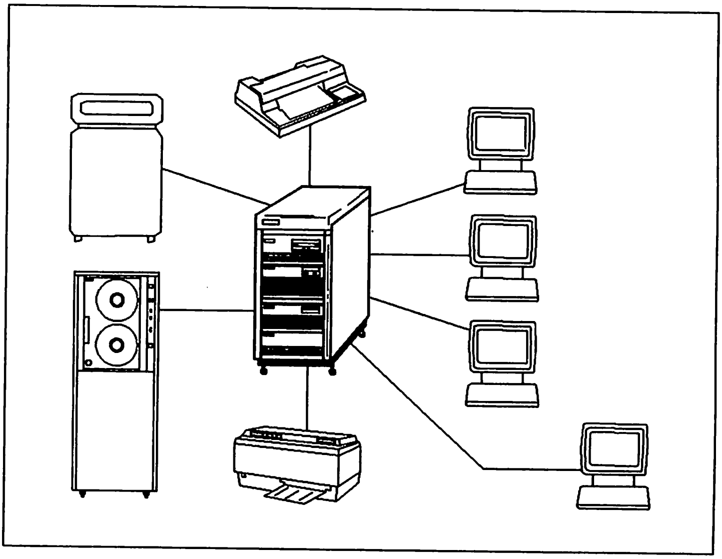


Figure 1. Terminal-Based Workgroup

This is a collection of terminals (or PCs with terminal emulation software) and peripherals connected to a host computer in a central location. Typically, the mainframe is in a special room, called a data center. Specific people, the Management Information Systems (MIS) group, are responsible for its control and operation. In this environment, users are only given access to and control over information they require. This scenario highlights a very centralized solution to a workgroup's computing needs. It allows for easy control and management of the system. Another advantage of this type of workgroup configuration is the accessibility to a very wide variety of peripherals and applications.

The problems with this type of solution are that bottlenecks can occur either with the system itself or the MIS team. For example, the performance and throughput of the system is inversely proportional to the number of users logged on and the type of applications they are running. Another example is that anytime a special operation must be done (e.g., restoration of an old file from a tape archive), the user must go through the operations staff. Occasional special requests are tolerable and not too costly to productivity. But, if they are frequent and there is only one operator to handle them, they can get be very time consuming and resource intensive. This potential for bottlenecks that eat up resources constitute one of the key disadvantages of this configuration.

2. The next figure illustrates a more decentralized workgroup setup.

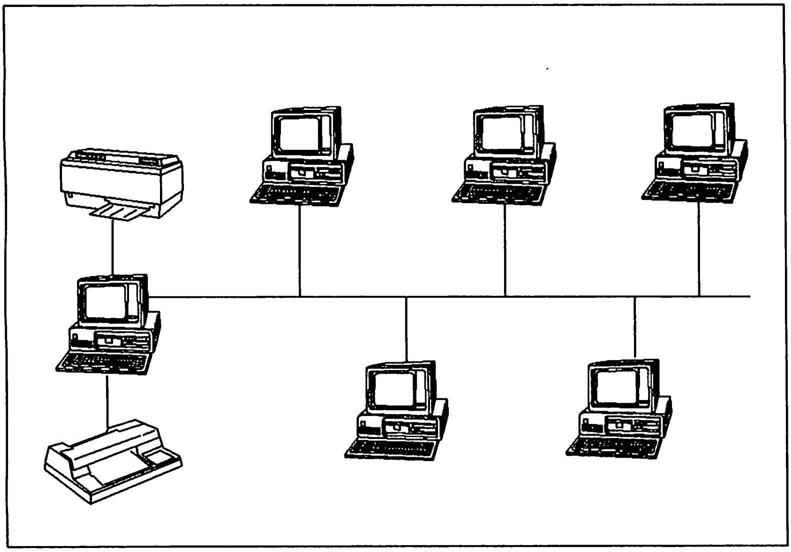


Figure 2. PC LAN-Based Workgroup

This configuration is a group of PCs connected via a Local Area Network (LAN) with one of the PCs dedicated as the server on the network. Each PC has access not only to its own software and peripherals, but also those configured on the server PC. Unless otherwise defined, each user is responsible for his/her own PC. A very decentralized solution like this alleviates many of the bottlenecks experienced in a centralized workgroup computing environment. In smaller companies or companies with limited data processing needs, a PC network is fine. As the company's processing needs grow, so does the size and number of PC networks. As these networks expand, it becomes more and more difficult for an MIS group to ensure the compatibility, integrity, and security of the software on them. The MIS group must then assume the roll of "LAN monitors" to verify that these points are not overlooked or forgotten. The key advantages of this configuration are performance and independence. Since most of the resources are localized to each PC, bottlenecks do not often occur. Disadvantages are the loss of control and the limited variety of peripherals that are supported on PCs.

3. This figure shows a combination of the first two configurations.

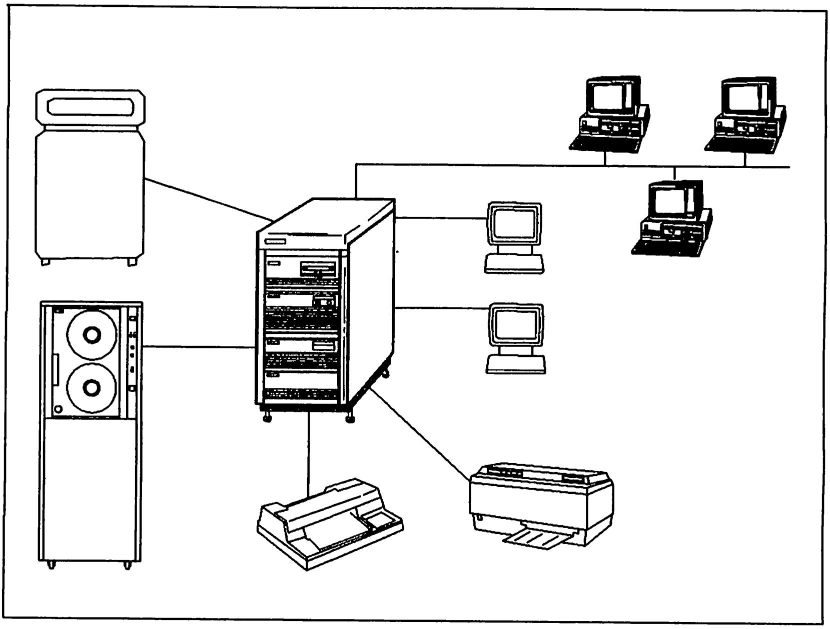


Figure 3. LAN-Based Workgroup with an HP 3000 server.

In this setup, both worlds are combined for maximum benefit. Most of the individual user's processing is still done locally on the PCs. The results can then be output via the host. (Note, 'output' refers to anything from sending a file to a printer to distributing it via electronic mail.) Be aware that the benefits realized are a function of the compatibility of the host-based server software and the PC-based server software. If the two co-exist well, the users can take advantage of both servers. This is the ideal solution in most cases. It takes advantages of the localized processing power of the PC, the connectability and processing power of the host.

III. METHODS FOR SHARING RESOURCES

Having reviewed some of the different configurations for an office computing environment, let's look at some of the methods for sharing resources. There are two primary areas where computing resources are shared. They are discs/files and printers. In either case, there are instances where it makes more sense to have several users share the same resource. One example is a very large data file containing a list of employee names and their extensions, where everyone needs to have easy on-line access to it. Having several copies of a file like this would be a nightmare to try to keep all copies up to date. Hence, there is a need for a single copy on a disc where everyone can access it and make changes as necessary. A different example is a marketing department that only has one letter-quality printer that also prints graphics. Everyone has a need to occasionally print to it, but this is hardly justification to buy several more. If the printer is centrally located and everyone can send their output to it, the problem is solved.

DISC/FILE SHARING

For a closer look at disc/file sharing, refer to Figure 4. Figure 4 illustrates two of the less automated methods for disc/file sharing.

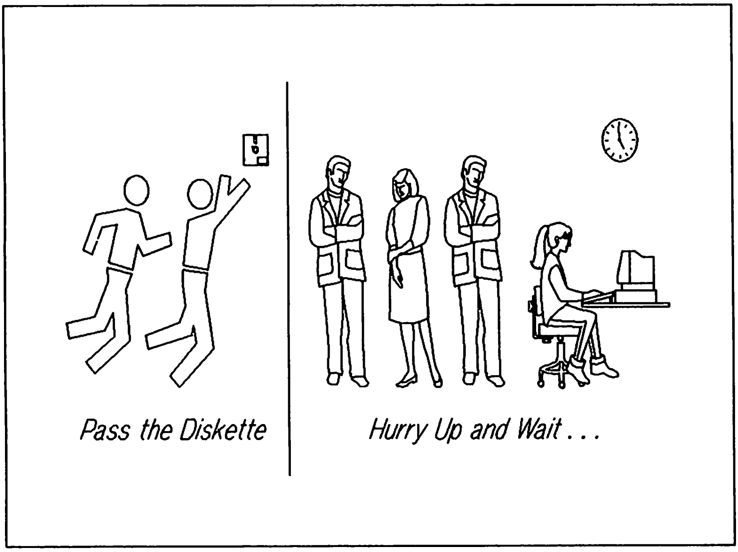


Figure 4. Manual Disc/File Sharing

In one instance, the file is stored on flexible disc and the disc is passed from user to user. In the other, the file is kept on a single PC and everyone goes to that PC to access it. In both of these cases, there are several opportunities for disaster to strike. One example of a disaster would be if the flexible disc got lost or damaged. Although Figure 4 shows two of the simpler forms of file sharing, Figure 5 illustrates three of the electronic techniques for disc/file sharing.

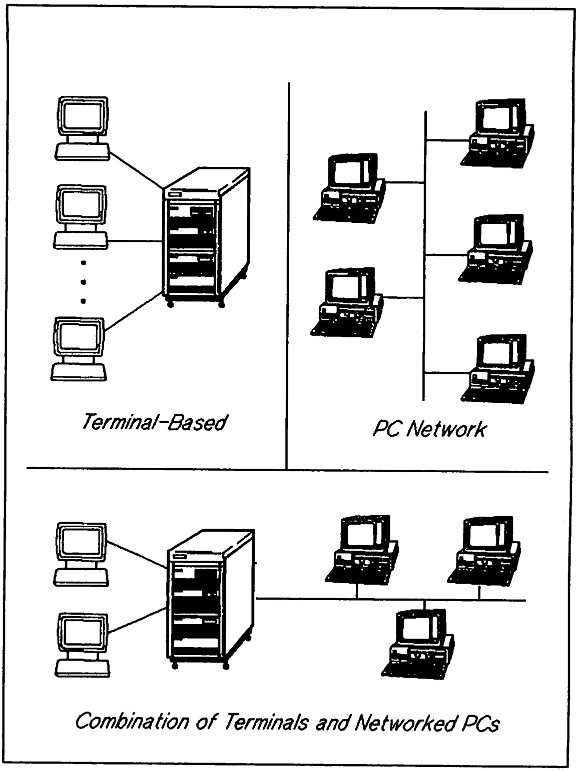


Figure 5. Electronic Sharing of Discs/Files

In the terminal/host based solution, files that several users may need are stored in a public group and account on the system. Depending on the level of security installed, users can display, copy or edit the files.

On the PC network, the file is kept on the server and then anyone on the network can access it. A drawback to this scenario is the physical limit on the amount of disc space that is available on PCs.

In the third exaple, the files are stored on the host system. Depending on the type of software that the server is running, these files may be in DOS format or that of the host's file system.

Another area of concern that is not directly linked to the sharing of discs/files is the protection and recoverability of data. This refers to the backing up and restoring of a user's PC disc. As for independent workstations or even those on a PC-based LAN, there is no simple solution to this problem. The two basic alternatives are to backup a PC's hard disc to flexible discs, or if available, to a tape drive. The latter solution usually requires additional hardware and drivers for the

PC to be able to communicate with the tape drive. This can be an expensive alternative to using flexible discs. On the other hand, the thought of backing up a 40 MB hard disc to a box of 360 KB flexible discs is also not very appealing.

PRINTER SHARING

Over the years several different methods have been developed for shared printing. In Figure 6, the two of the more basic techniques are shown.

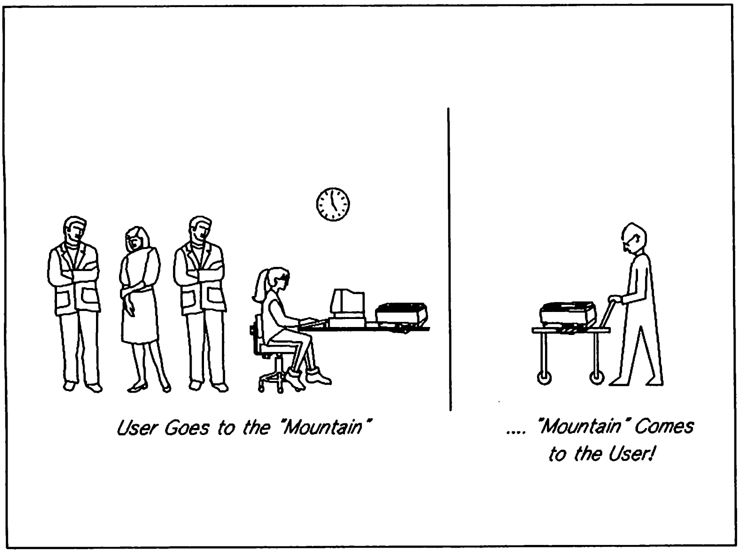


Figure 6. Early Shared Printing

In the first example, there is one printer attached to a PC. The users must bring their files to the printing workstation to get hardcopies. Although this is very cost effective from the standpoints of hardware and initial cost to implement, this can be quite slow and result in a queue of impatient users. This in turn translates into wasted resources and hence, a reduction in the return on investment over the long run. The other method involves rolling the printer from one workstation to the next and connecting it to the PC. Both alternatives can cause bottlenecks which means a loss of productivity.

Another print sharing alternative is shown in Figure 7.

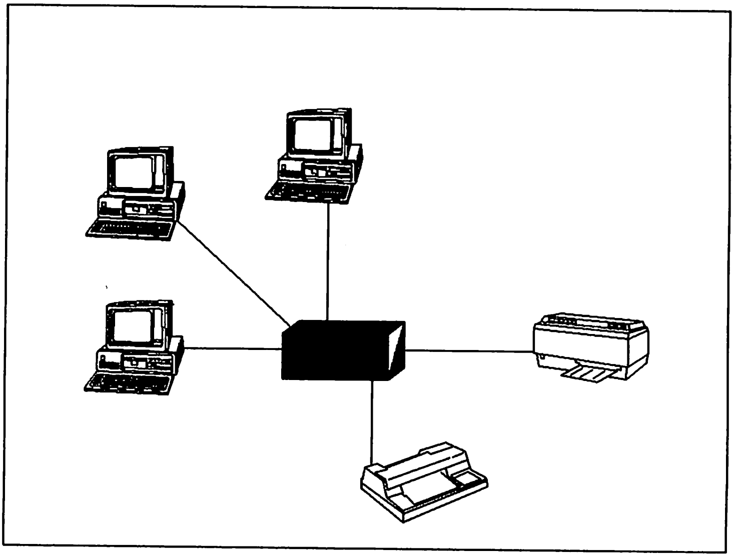


Figure 7. "Black box"/Switch Box

In this configuration, multiple workstations are connected to a printer via either a switch box or a "black box." Using a switch box, the users must manually switch the connection between the workstations and the printer. A "black box" is a more sophisticated switch box in that the switching is done electronically. In some configurations, the "black box" also allows spooling of output files. This feature is not available with switch boxes. Both are low cost solutions with the switch box being the least expensive.

Another approach to shared printing is illustrated in Figure 8.

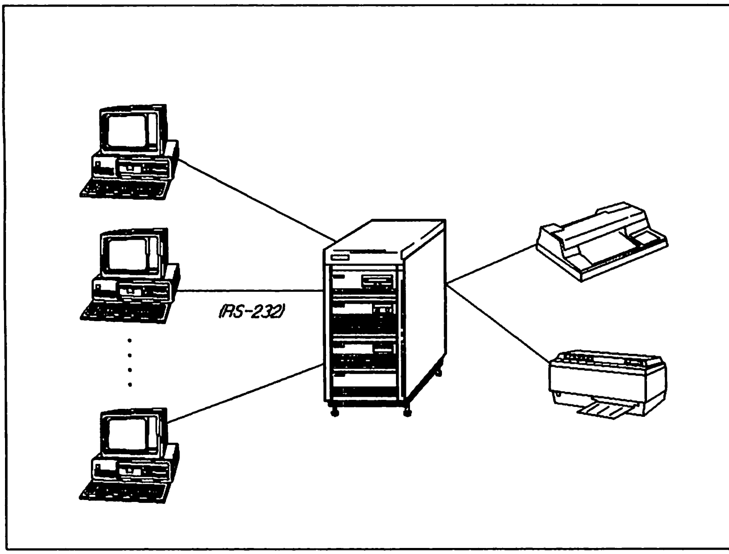


Figure 8. AdvancePrint

This scenario uses software that runs on the host system and the PC. The software provides a low cost interface for shared printing over a serial/RS-232 connection. Print files from the PC are transparently sent to the host system for output on one of the host printers. Using this software allows the PC user to send their output to the host taking advantage of the host's print spooling capability. Once their output is transferred to the spooler, their workstation is freed up for other tasks.

If a user does not have a serial connection to the host computer, but is connected to a PC server via a LAN, they can use the printers that are configured on the server. This also provides them with spooling, so their workstation is not chained to their printout. One limitation in this environment is the limited variety of printers supported on PCs.

IV. ONE EXAMPLE OF A POSSIBLE SOLUTION

For this example, let's look at the situation where the workgroup needs the performance and independence of a PC network but does not want to sacrifice the flexibility and control of a host-based solution. Referring to the second section of this paper, where three different office system configurations were given, the best solution for these users appears to be the combination of a PC network with a mainframe host as a server. Remember, the degree to which the combination is the best solution is dependent on the compatibility of the host-based networking software and the PC-based networking software. If the two co-exist well, then the users truly realize the maximum benefit of both environments. An example of this type of package for an HP 3000 host system is a product called Resource Sharing.

Resource Sharing runs on the HP 3000 and is designed to co-exist with PC servers on a network. It adheres to the AdvanceNet and MS-NET specifications. This means that an HP 3000 server running Resource Sharing can be added to an exising PC-based network without any problems. Figure 9 illustrates this environment.

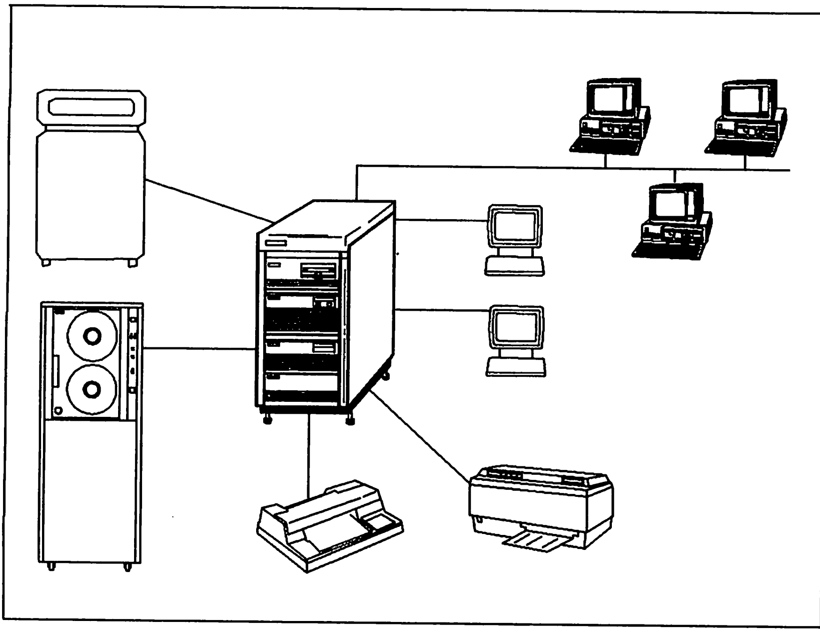


Figure 9. Network of PCs with an HP 3000 Server

In this configuration, Resource Sharing allows terminals to continue to function as though it was a stand-alone system. PC users can connect to the HP 3000 and log on as terminals for running HP 3000 applications. But, more realistically, the PC user can do all processing on the PC and if necessary, can easily move the results to the host. Once on the host, the information can be distributed by any number of means, including electronic mail.

From a PC workstation, a user can create a shared disc on the host. A shared disc appears to DOS like any other PC disc, except that it is created on an HP 3000 disc drive. In brief, it overlays the MPE file structure with the DOS structure. This results in one shared disc being created in one MPE group. As a result of this, the limit on the number of files that can be stored in a shared disc is a function of the maximum number of files that can be stored in an MPE group. On the same note, the maximum size a shared disc can be is determined by the disc space allocation for that

MPE group. As long as a user can log on to a group they can create a shared disc in that group. The user can also allow other users to access their shared disc, making file sharing much more convenient. The user can also put a password on the shared disc preventing unauthorized access to it. Connections to shared discs can be done either dynamically or when the user reboots their PC as part of the network software loading process. There is also a utility that allows the user to backup and recover shared disc files. This is an added safeguard, since the shared disc files are backed up as part of a normal MPE SYSDUMP/STORE operation. This utility also has the added flexibility to recover shared disc files from standard MPE SYSDUMP/STORE format backup tapes.

Aside from the utility to backup and recover shared disc files, a PC user can also backup and recover their local hard discs to/from a tape drive on the HP 3000. The PC BACKUP utility also allows the system manager to schedule backups, so as to not create bottlenecks in the input/output subsystem of the HP 3000.

Another feature of Resource Sharing is shared printing. A user can redirect the output from their PC to almost any printer on the HP 3000 (as long as it's configured in Resource Sharing). This includes printing text and graphics. It is also possible to plot from a PC to a spooled plotter on the host. Once in the MPE spooler, the output can be manipulated in the same way as any other HP 3000 spooler file. In Resource Sharing, the system manager can customize the printer configuration to maximize the effeciency of printing. This significantly helps to eliminate bottlenecks when printing documents.

Resource Sharing also has a utility that allows the user to do file conversions from DOS files on a shared disc to MPE files. This allows the users much greater flexibility in manipulating data if a PC is not always available. In other words, a user could be working on a file in the office on their PC and store it on one of their shared discs. Then later that evening while at home, they could dial in over a modem with a terminal, convert the file to MPE format and edit it using an MPE editor. Then, once they are complete, they could convert the file back again to DOS format.

V. CONCLUSION

In closing, it was not and is not the intention of this paper to intentionally convince you that one configuration is better than another. Hopefully, you are now aware that there are still steps that you can take to optimize the productivity of your system and its users. As with just about everything else, balance is necessary in designing, implementing, and/or upgrading a computer workgroup. Achieving good balance between a centralized and decentralized solution will afford you good control and maintainability without significantly sacrificing performance. The purpose of this paper is to create an awareness of some of the opportunities available to you to increase productivity through resource sharing!

Effectiveness vs. Efficiency in Managing a Large, Distributed EM Network

by

Luis Hurtado-Sanchez and Amy Tada Mueller[1]

Hewlett-Packard Co.
3000 Hanover St.
Palo Alto, CA 94304

I. Introduction:

A search of the literature over the last three years revealed little information on the determinants of cost for electronic mail (EM) networks of whatever size and little guidance offered organizations on how to manage their EM network costs. This paper aims to partially fill both gaps. It should prove particularly useful to current or future implementors of EM networks using HP DeskManager (HP Desk), Hewlett-Packard's (HP) EM product based on the HP 3000 line of computers.

HP's internal HP Desk network is the source of the data presented and discussed in this paper. Implementation of this network began early in 1982. The data itself was gathered in the Spring of 1987. As of July 1987, HP's internal HP Desk network had over 63,300 users registered in the network's global directory. The network connected 536 HP 3000's in the United States and 32 other countries. Average volume per month through the network was

1. Luis Hurtado-Sanchez is Integrated Office Systems manager and Amy Tada Mueller is HP Desk Messaging Section manager at Hewlett-Packard's Corporate Offices in Palo Alto, California.

calculated to be 24 billion bytes, or approximately 4 million pages (with an average page estimated to contain 4,000 bytes).

The HP EM network is designed as a tiered network with different levels of service and delivery times. Average delivery time anywhere in the world is seven hours using normal service and one hour using urgent service. Local delivery time is usually in minutes. Table 1 summarizes the major facts about the network as of July 1987.

Data was published in a Spring 1987 issue of the San Francisco Examiner newspaper regarding the number of subscribers of the major public EM networks in the United States. Using that data, HP's EM network would rank between the seventh and eighth largest public EM networks in the U.S. in terms of registered users.

More information regarding the origin and development of HP's EM network is contained in an article in the September 1986 issue of the HP Journal.[2]

II. Organization of This Paper:

This paper begins by analyzing the nature and determinant of costs in large, distributed EM networks, such as HP's. Next, it examines how such costs and the factors affecting them could vary in comparison with smaller networks or with non-distributed networks. The first two sections of the paper lead to the next section, where recommendations are made regarding cost management for EM

2. "Implementing a Worldwide Electronic Mail System," Luis Hurtado-Sanchez, Amy Tada Mueller, et. al., Hewlett-Packard Journal, September 1986, pp. 30-48.

networks. Lastly, cost comparisons are made with other alternative methods of communication, both electronic and not.

III. Nature and Determinants of Large, Distributed EM Network Costs:

Total costs of operating HP's EM network were calculated to be approximately $ 3.7 million per month ($ 44.4 million per year). This equals $ 58 per month per registered user. (See Tables 2 and 3.) All costs presented in this study are current costs, that is to say, they are the costs of currently running the network. They reflect no purchase or acquisition costs.

There are three major components to the HP EM network costs: 1) current operating costs; 2) human resources costs; and 3) data communications costs.

Current operating costs make up 80.2 percent of total costs. Under the current operating costs heading are the costs associated with running the HP 3000's on which HP Desk itself runs. These costs include depreciation, service and support, operations, occupancy, and the like. A standard operating cost was calculated for each HP 3000 series model in use in the network. This cost was then applied to all the HP 3000's in the network dedicated to running HP Desk. For those HP 3000's not dedicated to running HP Desk, a percentage of the standard operating cost was applied; this percentage was the equivalent of the percentage of each HP 3000's resources calculated to be devoted to HP Desk. Current operating costs of the network do not include the cost of personal computers (PC's) and terminals used to access the network. There are two main reasons for not doing so. First, many, even most, of the PC's and terminals are fully depreciated or were fully expensed when acquired. (Costs included in this study are current costs, not reflecting purchase or acquisition costs.) Second, the PC's and terminals accessing the network were originally procured to do other tasks, such as using other EDP systems, doing office automation, and so on. Thus, their contribution to the current operating costs of the network was taken to be minimal.

Human resources costs make up 12.1 percent of total costs. These costs are the costs associated with the people who support the software and train and support users internally. They are called local messaging coordinators (LMC's) and are located at major entities (manufacturing divisions or sales regions) throughout HP. Human resources costs include salaries, benefits, travel, training, occupancy, and the like. (Thanks are due the LMC's for their time and effort in gathering all the data used in the cost study.)

Data communications costs make up 7.7 percent of total HP Desk network costs. These are the costs of transmitting the data between (not within) entities through HP's private data communications network. Data communications costs for EM within an entity were taken to be minimal, since such a network is used by many applications and for many purposes, of which EM is only one. Also, no billing or similar measurement mechanism existed to estimate the local networking costs due to EM. HP's inter-entity data communications costs are probably low compared with those of other organizations, since HP moves its EM through its own private data communications network. This network consists of a combination of leased point-to-point lines and switchable X.25 circuits. The rate used to calculate data communications costs is $ 35 per megabyte, a weighted average of U.S. and international costs.

As far as volume itself is concerned, only inter-entity volume was actually measured. Experience shows that intra-entity volume is approximately twice inter-entity volume. Thus, total volume is calculated to be three times the measured inter-entity volume.

The cost per page can be calculated from the data given in terms of monthly volume and total cost. The average cost per page is $ .61. However, HP Desk possesses several features which allow a single copy of a message transmitted through the network to serve several users. Since the average number of recipients per message is 2.075 (based on a sample), the average cost per page per recipient is $ .29. (See Table 4.)

Regarding HP's EM network costs and how they compare with what HP's customers might experience, it is true that HP obtains its hardware and software internally cheaper than customers. However, as noted previously, the computed EM network costs are current costs and do not include purchase or acquisition costs. They do include an average estimated depreciation for the HP 3000's. Other costs, such as salaries and benefits for the LMC's, occupancy, and leased lines, may be the same as, higher, or lower for HP than for its customers. An analysis of the data leads us to conclude that a customer's comparable EM network costs, computed under similar assumptions, would be higher than HP's by no more than 15 % and in many cases might be the same or lower.

Table 5 summarizes the results of regression analyses conducted on several variables to determine predictive relationships between several independent variables (number of users per entity, number of computers per entity, number of users per computer, outgoing volume, and others) and several dependent cost variables. A stepwise (up) procedure was conducted to select the best fit predictive model with the least number of independent variables.

The major conclusions to be drawn from the regression analyses are that at the level of an entity (manufacturing division or sales region):

1. HP Desk network current operating costs are driven largely by the number of users and the number of computers.

2. HP Desk human resources costs are also driven largely by the number of users and the number of computers.

3. But, the number of computers which belong to the network is largely driven by the number of users.

IV. How Costs Might Differ in Other Types of EM Networks:

The analysis in the previous section of this paper was conducted largely in absolute terms, from data for a very specific kind of network, since HP's internal EM network is both large (536 HP 3000's, accessed by approximately 30,000 PC's and 30,000 terminals) and distributed (the HP 3000's are situated in the United States and 32 other countries). The question arises of how the data and therefore the analysis would differ for different kinds of EM networks, or in different kinds of organizations from HP. This section will speculate on these issues.

First, consider smaller networks. We believe that for smaller but still distributed networks, the total costs would be comparatively smaller. The percentage distribution of costs into the three different categories considered would be similar. However, human resources costs would be a larger percentage of total costs because of the fixed overhead associated with administering HP Desk on each HP 3000. Such overhead does not decrease proportionately as the size of the network is reduced. Although total costs would be smaller for smaller networks, we believe they would not decrease sufficiently to allow for an even larger decrease in volume. Thus, we would expect unit costs to be higher for the smaller, distributed networks.

For equal-sized but centralized networks, the costs might be larger or smaller, both in total and on a per unit basis. The distribution of costs, however, would certainly be different. The total computer resources needed would be smaller due to sharing system overhead among a greater number of users on larger computers. How many fewer computers would make up the network depends on the economies of scale effected by consolidation. Fewer computers would also lead to smaller costs for human resources to support the network and its users. However, data communications costs would be higher, perhaps much higher, as users would have to access the computers from remote sites. How much higher would depend on several factors, including how far users would be from the computers they would have to access and the costs of the network they would use to access the computers. Centralized networks may also have other, not easily quantified costs, such as less control by end-using organizations and less integration, if users must use EM on a computer other than their home computer (the one on which they usually do the bulk of their daily work).

To summarize, centralized networks would have lower operating and human resources costs than decentralized networks. However, decentralized networks would have lower data communications costs and probably greater integration and control by the using organization, leading to greater productivity. The challenge for any organization is to arrive at that optimum point where these considerations balance each other. For example, in HP's EM network, approximately 50 % of the HP 3000's have HP Desk installed on them. Thus, even though HP's network is highly decentralized, it is not as totally decentralized as it might theoretically be.

V. Recommendations on Managing EM Network Costs:

Before making recommendations on management of EM network costs, the data collected needs to be analyzed in more detail. Table 6 breaks down the data by looking at monthly per registered user costs within categories defined by the number of registered users which use a given HP 3000 as their home computer.

At first glance, the data suggests that the way to reduce EM network costs is to put as many users on an HP 3000 as possible. However, the analysis so far does not take into account the intensity with which certain users may be using the EM network. For a more refined analysis, a measure of the intensity of use of EM on an entity-wide basis was defined:

$$\frac{\text{Avg. disc space/user/entity}}{\text{HP-wide avg. disc space/user}} + \frac{\text{Avg. outgoing vol./user/entity}}{\text{HP-wide avg. outgoing vol./user}}$$

Table 7 displays four different kinds of normalized data within categories defined by intensity of use. Average cost per user per month goes down with an increasing number of users per computer. However, once the average cost per user per month is divided by the intensity of use, the normalized average cost per user per month goes down with decreasing number of users per computer.

The data thus really indicates that the key cost management decision to be made is balancing the allocation of resources to meet the varying usage patterns. Thus, organizations with high intensity of usage should insure that users are making proper use of EM. In terms of cost management, they should aim for low unit costs. Organizations with low intensity of usage should examine whether their policies and procedures are keeping usage, and thus, effectiveness, artificially low. In terms of cost management, they should aim for low total costs. Both types of organizations should implement some kind of billing system, at least to educate users. Billing should be usage-based and structured around the resources in shortest supply.

VI. Comparisons with Alternatives:

Although the data gathered on the costs of HP's EM network is both informative and useful on its own, it is even more so when it is compared to similarly gathered costs for alternative forms of mailing and communications, both electronic and not.

Table 8 presents comparative cost data for several alternatives to HP's EM network. Table 9 presents comparative delivery times data for the same alternatives.

The inter-office mail, HP Desk network, and FAX costs are based on data internal to HP. So are the Comgrams costs (Comgrams was an earlier, TELEX-like messaging system broadly in use within HP.) Public E-Mail networks referred to are two large public electronic mail networks within the United States. Express Mail includes the average cost of the service and an allocation of HP's internal costs in handling it.

VII. Conclusions:

From Tables 8 and 9, it can be concluded that an HP Desk network comes closest to being the ideal means for normal and urgent communications, when both cost and speed of alternatives are considered, for short to medium length transmissions. For deferrable communications, when a delivery delay of several days is permissible, regular mailing channels (inter-office mail and the postal service) are both cheaper than EM. For very long communications, paper mailings may be preferable in terms of cost and convenience to the receiver (after all, who reads a 50-page newsletter online?). However, EM may prove more convenient to the sender for long communications.

Several challenges lie ahead in managing HP's internal HP Desk network over the next few years. HP's EM network is now well established. It has proven to be a competitive advantage from the standpoint of sheer communications cost and speed. It is used to send not only messages but also documents, graphics, spreadsheets, source code, object code, newsletters, almost anything which can be encoded electronically. It is used to manage projects across entities, across time zones, across the world. In effect, it has become the <u>personal information distribution system</u> of HP, no longer just a messaging system.

Opportunities await to turn HP's EM network into an even greater competitive advantage from the standpoint of other business considerations. One such opportunity lies in promoting greater integration between EDP applications and the EM network, adding to its current role as the personal information distribution system of HP. Another opportunity lies in hooking up HP's EM network with similar networks that HP customers and suppliers enjoy. This interlinking of EM networks would result in greater responsiveness to issues and faster problem resolution. There is also the challenge of ever increasing volume caused by increased demands on the network stemming from increased use of EM, greater user sophistication, and technology evolution. In particular, the network will have to change to meet the future challenge of increasing message size brought about by such technologies as desktop publishing, image processing, and digitized voice.

Arching over the mission to satisfy users' needs and the desire to take advantage of new opportunities and challenges is the mandate to reconcile cost and service considerations to provide HP with an EM network which is simultaneously effective and efficient. In the world of electronic mail networking, there is only one law: Follow up, fine tune, forever.

TABLE 1 **HP's HP Desk Network**

July 1987

SCOPE

33 Countries

536 HP 3000's

63,386 Registered HP Desk Users

VOLUME

24 billion characters/month

(6 million pages/month)

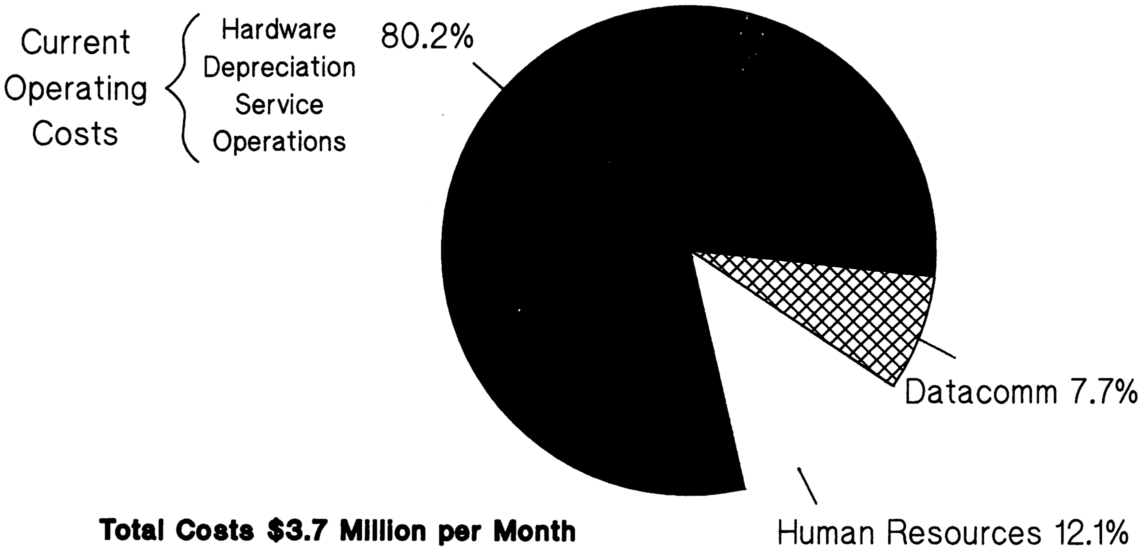
DELIVERY

Avg	Range	
10 sec	0–5 min	Within an HP 3000
30 min	0–2 hr	Within a location
1 hr	0–2 hr	Between locations (urgent)
7 hr	0–12 hr	Between locations (normal)

STAT8707

2014-13

TABLE 2
Mid - 1987 HP's HP Desk Network Costs



Total Costs $3.7 Million per Month

COSTPIE

2014-14

TABLE 3

Mid – 1987 HP's HP Desk Network Costs
Average Cost per User

Total HP Desk Costs	3.7	($M/month)
# of Registered Users	63,386 *	
Average Cost per User	58	($/month)

* Registered in HP global directory

AVGPU

2014-15

TABLE 4

Mid – 1987 HP's HP Desk Network Costs
Average Cost per Page

Total volume (per address)	24,168	(Mbytes/mo)
Total volume (per recipient)[1]	50,149	(Mbytes/mo)
Total cost	3.7	($M/mo)
Cost per page (per address)[2]	.61	($)
Cost per page (per recipient)[2]	.29	($)

[1] estimated 2.075 recipients per address based on sample
[2] assumed 4000 bytes per page

AVGPP

2014-16

TABLE 5

Mid - 1987 HP's HP Desk Network Costs

Cost Regression Analyses

$Y = a + bX_1 + cX_2$	R^2	a	b/std err	c/std err
Y = Operating costs ($K/mo) X_1 = # of users X_2 = # of computers	.78	−3.9	.012/.004	4.72/.44
Y = Human resources time (100 = one person) X_1 = # of users X_2 = # of computers	.60	36.6	.059/.008	2.05/1.00
Y = # of computers X_1 = # of users	.50	1.4	.006/.0006	

REGSUM

2014-17

TABLE 6

Mid - 1987 HP's HP Desk Network Costs

Cost Analysis by Users per Computer

# Users per Comp	Avg $ per User per Month
20 - 50	107
51 - 100	62
101 - 200	40
201 - 300	30
300+	22

ANALUPC

2014-18

TABLE 7

Mid – 1987 HP's HP Desk Network Costs

HP Desk Cost Analysis by Intensity of Use

Intensity of Use (Avg = 2.00)	Avg # Users per Comp	Avg $ per User per Month	Avg $ per User per Month/ Intensity
0.00 – 1.00	258	34	47
1.01 – 2.00	143	49	35
2.01 – 3.00	98	75	29
3.01 – 9.49	77	130	29

ANALINT

2014-19

TABLE 9

Mid – 1987 HP's HP Desk Network Costs

Comparison with Other Alternatives

($ Cost per Page per Recipient)

System	2 Pages	10 Pages	Delivery
HP Interoffice Mail	.10	.50	1–5 days US, 5–10 days Intl
US Postal Service	.44	.88	1–3 days US, 3–10 days Intl
HP's HP Desk Network	.58	2.90	1 hr urgent, 7 hrs normal
Comgrams	.76	3.80	1–3 days
Public E–Mail Net #1	1.60	4.80	Varies
Public E–Mail Net #2	2.00	6.00	Varies
FAX	3.20	9.68	Varies
Express Mail	10.00	10.00	Overnight

Note: Does not include costs associated with message preparation

2014-20

COMPDEL

Utilizing the Personal Computer for MPE Performance Management

Rex Backman
Hewlett-Packard Company
Roseville, California

Introduction

Coupling the personal computer as the presentation manager with a host based MPE mainframe data collector is the basic infrastructure found in a powerful and user friendly performance management solution available now from HP. The product LaserRX, provides the local System Manager or Performance Specialist the ability to view in a full color, graphics based, MS Windows environment the performance information collected on the MPE host over a predetermined user defined time period. The ability for the user to define what metrics to collect, when to collect, and how to display (by hour, week, or month) provides a limitless amount of performance metrics. LaserRX's flexibility allows the user to view the performance issues that they feel are pertinent to their site in a time frame they feel is appropriate.

Having used the product over the past few months I'd like to share the positive experiences we have had with LaserRX. A very brief overview will provide a basic understanding of the product structure. Examples of the displays that LaserRX provides will be shown along with examples and ideas such that System Managers can see for themselves the potential of LaserRX in the area of performance management.

Product Structure

Prior to LaserRX, I as a System Manager of a multi-CPU HP3000 shop had available to me an incomplete set of performance tools and services. OPT/3000 provided me with the immediate "what's going on now..." information needed in periods of reactive system tuning. On the other end of the spectrum, there was HPTrend and its' ability to show "where we have been.." albeit in a rather non-flexible fashion. The missing link was the ability to view in the local environment the time periods found between the immediacy of OPT reports and the long term perspectives of HPTrend. From my experience LaserRX fills this void with amazing functionality and flexibility.The System Manager can determine when to collect data, what data to collect, and most importantly how to view the data using LaserRX's MS Windows full color graphics capability. Performance metrics such as memory bottlenecks, CPU utilization, Disc Activity are available with ease from LaserRX. In addition, LaserRX allows customization of local environment classes. Classes are easily defined by the System Manager and allow for a site to determine who their CPU consumers are. Examples that I have used for classes are HPDeskmanager, Print Spooling, plus several unique application systems (Cost Accounting, Order Processing, etc). Each class mentioned here was defined based on the fact that it was deemed important to our shop, we wanted to get a handle on how these classes were effecting our system. LaserRX allowed us to do this task easily and effectively. Thus our site specific classes coupled with global performance displays allows us to get a complete picture of how our machine is performing during time periods that we deem important.

The structure of LaserRX is a two tiered design, resident on the MPE V/E based machine is the data collector. This batch job, a low CPU consumer, continuously gathers the pertinent (defined by the user! - remember we collect what we think is important) performance metrics. Three log files residing in the SYS account are the sole areas of data. No other resources are used for data storage. Your system log files and network log files remain untouched. Log files used by the host data collector are of a circular design with the size determined by the System Manager. This allows for flexible data management of the performance metrics collected on the host. Local sites can size their log files to contain 2 days, 2 weeks, or 2 months or whatever depending on their reporting needs and available disc space resources.

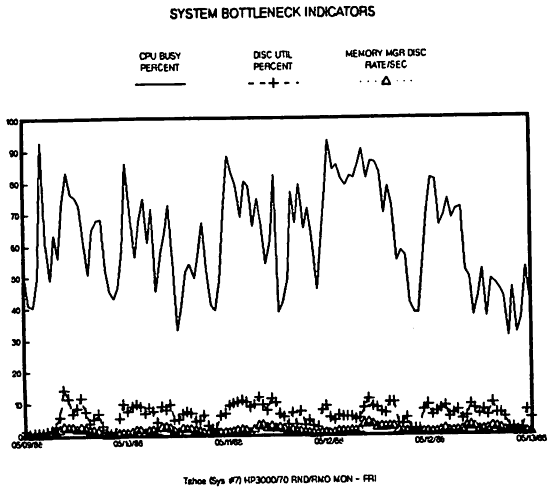
The complementing side of the architecture is the PC. On an HP Vectra or Vectra compatible, the requirements are: MS Windows 2.0, MS-DOS 3.1 or later, and a datacomm link (serial or 802.3). The workstation platform is the presentation engine for the host based collected data. Full color graphs on CPU bottlenecks, Transaction Response time, CPU utilization, Disc Activity plus user defined classes can be displayed on demand with a few easy "clicks" of your Mouse.

Obviously, the merger of the PC with its' sophisticated display environment coupled with an intelligent and flexible data collector provide limitless ways to view performance management issues. System Managers now have a solution to see where they have been, who their consumers are, and with LaserRX's ability to export data to other PC packages, a mechanism to model the raw data to fit their reporting needs.

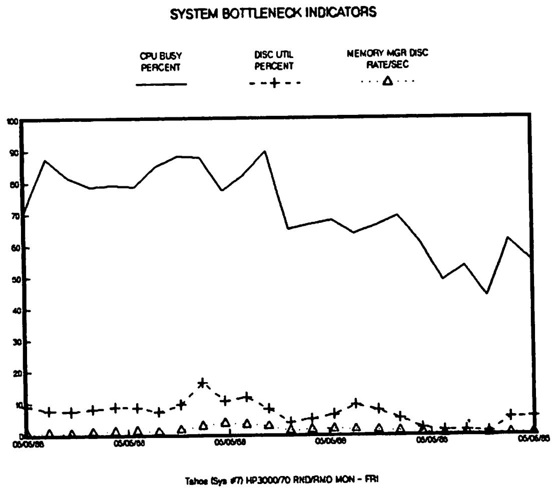
The amount of raw data is truly staggering. While this could be thought of as a negative point I view it as one of the most positive features inherent in LaserRX. Truly, only a day-to-day member of a site can grasp what data (performance) metrics are important. LaserRX lets you choose! Areas that can be displayed/printed from the PC based Windows environment are:

CPU Bottlenecks
CPU Utilization
Disc Transactions
Transaction Response Time
User defined Classes

Below are some examples of "traditional" system performance graphs. All of these graphs were created with the same data points, the time focus and class descriptor were the modifying parameters. While it is hard to mimic the PC color based graphics environment in a black and white paper such as this, the graphs do represent an example of the power of LaserRX.



LaserRX Example: Five Day System Bottleneck Graph



LaserRX Example: One Day System Bottleneck Graph

In addition to the traditional performance measurements, LaserRX also graphs the user defined classes. The classes are up to the creativity and need of the local System Manager. Definitions for the user defined classes as well as the triggers for the generic or traditional measurements are resident in the file "SCOPPARM". This parameter file is configured by the local System Manager in a very forgiving syntax. SCOPPARM has two basic components, the first is the traditional system workload or "global" metrics. The second component of the file contains the locally designed class definitions. An example of a SCOPPARM file used in our shop on a HP3000 S/70 is shown below:

LaserRX Example: Host Mainframe Parameter File

```
ID Tahoe (Sys #7) HP3000/70 RND/RMO
*************************************************************************
            **>>Global CPU LaserRX Metrics<<**
*************************************************************************

** Log all three types of records with log command **

LOG GLOBAL WORKLOAD PROCESS

** Threshold interesting processes are ones that: used 20% of the CPU, or  **
** had >10 I/O's per second or > 10 second response time, log these process**
** events to the LaserRX Process log file.                                 **

THRESHOLD CPU=20, DISC=10, RESPONSE=10.0

** Use default response time buckets **

RESPONSE RANGE= .5, 1, 2, 3, 4, 5, 10, 20, 40
FIRST    RANGE= .1, .2, .3, .4, .5, 1, 2, 3, 4
THINK    RANGE= 1, 5, 10, 20, 30, 40, 50, 60, 120

*************************************************************************
            **>>LaserRX Class Definitions<<**
*************************************************************************

** Class for performance tools **

CLASS=PERFORMANCE TOOLS
FILES=OPT.PUB.SYS,@.SCOPE.SYS

** Class for Datacomm **

CLASS=MPE DATACOMM
INTERACTIVE=DS@.@.SYS,@.NET.SYS
BATCH=DS@.@.SYS,@.NET.SYS

** Class for HPDesk **

CLASS=HPDESKMANAGER
FILES=@.HPMAIL.SYS,MAIL@.@.HPOFFICE

** Class for Maestro/Tapes **
```

CLASS=MAESTRO/TAPES
FILES=@.@.CCC

** Class for Print Spooling **

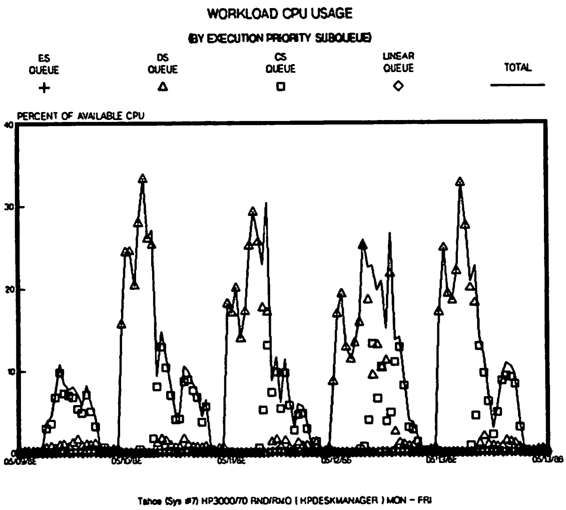
CLASS=PRINT SPOOLING
FILES=@.@.RSPOOL,@.PSPOOLER.SYS

** Class for PC software monitors **

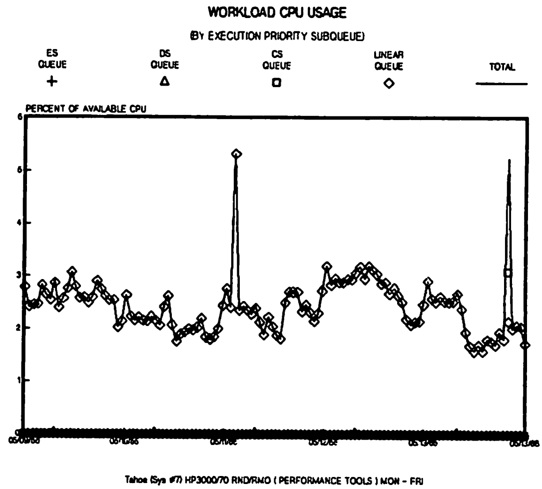
CLASS=PC SOFTWARE
FILES=MONITOR.PUB.SYS,@.PPC.SYS,@.PPCUTIL.HPOFFICE,PCLINK.PUB.SYS

Using this example of a SCOPPARM file we can now present some user defined class graphs. Class graphs is an area that makes LaserRX so powerful. The ability to determine who the system consumers are is up to the discretion of the System Manager. It is now a simple task to found out what these consumers did.

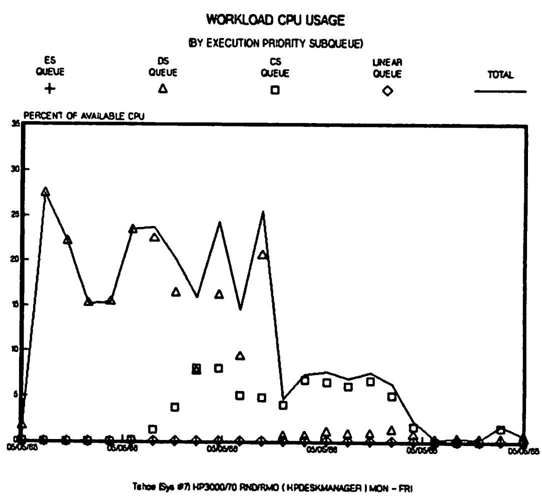
Now let's look at some more LaserRX graphs. Again these are not exactly as they will appear on a PC, but they are representative of the product:



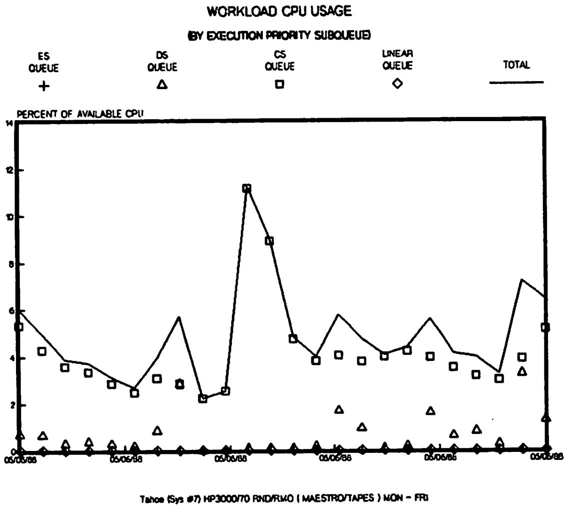
LaserRX Example: HPDeskmanager Five Day Workload Graph



LaserRX Example: Performance Tools Five Day Graph



LaserRX Example: HPDeskmanager One Day Graph



LaserRX Example: User Defined Class (Third Party Software) Graph

Potential

Now that we have discussed and presented a very brief overview of the product structure, lets turn our attention to what solutions the product can provide. Flexibility and freedom are the key phrases. Data collected, especially class metrics are user controlled. As a System Manager you have the capability to define what to collect and how to present the LaserRX data points. Resolution is up to the PC user. Hours, days, or weeks can be presented depending upon the resolution required. In our uses of LaserRX, we graph and display the last business week (Mon - Fri) in graphs of 24 hour days. This allows us to present to our user community an easy to read graphical based representation in full color of where our machines operated.

Each graph generated by LaserRX has its own useful and unique purpose. For example, the classic graph of CPU Utilization presents an overview in a stacked line chart manner showing the partitioning of the CPU as a resource. Interactive consumers, batch consumers, system processes, memory management, ICS overhead, and Paused for I/O are represented by a different color allowing for quick identification of problem areas. In analyzing the display, problem areas are easily identified by color and by large percentages of the overall time period. System Managers can also get a good idea of the distribution of interactive versus batch processing utilized on the machine from this display.

Transaction Response Times, long a desired feature in HP performance tools is available in LaserRX. Metrics in this display are: Time to first response, Average time to prompt, and Transaction rate (x 1000). The first two metrics are in seconds for interactive sessions. Transaction rates are the number of terminal transactions over time for the selected measurement period. Our use of these graphs has been to compare periods of high CPU use versus low periods to gauge the impact of a busy system in regards to user response time. Unexpectedly this graph also helped us to identify a performance bottleneck that, unknown to us, was happening in the early morning hours due to some hardware anomalies. Again the flexibility of the product is seen here. We were not aware of this problem, yet by graphing "where we had been" over a previous weeks data points we recognized and resolved a performance problem.

Class graphs are useful in further analyzing the partitioning of the interactive and batch applications resident on the machine. By properly defining classes, answers to questions like "How much is HPDeskmanager consuming on the machine?" are easily answered. Class data provides metrics that allow for System Managers and the Application Programming staffs to help evenly distribute system loads. Trends can be easily seen allowing for changes in application consumption to be easily identified.

PC Export Facility

Extended and unique functionality is found in LaserRX's Export facility. This portion of the product allows the PC user from their Windows session to "click" on a range of data points and to export them out of the LaserRX environment and into MS DOS data files. Once resident in these MS DOS data files they can be incorporated into several MS DOS packages. Spread sheets (LOTUS 1-2-3, Excel by Microsoft, etc) or pure graphics packages (Graphics Gallery) can be used. This export facility allows the LaserRX user to remove and isolate the performance metrics that they wish to examine in more detail out of the LaserRX environment.

At this point we have learned that LaserRX data moves easily and effortlessly from the HP3000 data collector, thru the LaserRX MS Windows environment and finally to a user defined MS DOS data file. All accomplished via clicks of the Mouse or a few easy keystrokes. Again, the data that is ported is truly determined by the System Manager. The LaserRX user determines what exactly to them is important to collect, to view, and to export!

Conclusion

As a System Manager with several machines to support, LaserRX has been an immediate and positive addition to my tool set. Performance metrics are easily gathered and displayed. I have the flexibility to define what metrics are gathered and when. In our shop LaserRX has become a transparent data collector that we use on a weekly basis. The product has allowed us to make intelligent choices on hardware issues, third party software evaluations, and the week-to-week performance displays we present to our user community.

Areas that before were best guesses will be areas of defined and represented measurements with the use of LaserRX. Capital budgeting decisions can now be based on exact needs and not feelings. Load balancing is easily attained, application tuning as well, are readily available results from LaserRX outputs. This tool, in its' first release, is a magnificent solution to help System Managers move from a reactive system resource utilization mindset to a proactive mindset. Whether your shop has one system or several, this product can make system resource measurement a proactive science. Once exposed to LaserRX, the ability to extract and analyze is in the hands of the user. A shop can now easily and quickly track where their system(s) have gone, allowing for knowledgeable and timely decisions on how they can prepare for the future.

User Interface Design Methodologies for CD-ROM Information Retrieval or "How to Find that Needle in a Haystack"

Greg Ferguson
Hewlett-Packard
Application Support Division
Mountain View, California

"There may be millions of fine thoughts, all encased within stone walls of acceptable architectural form; but if the scholar can get at only one a week by diligent search, his syntheses are not likely to keep up with the current scene."

Vannevar Bush
"As We May Think"
The Atlantic Monthly; July 1945

The preceeding quote describes a phenomenon that exists today, even with the vast amount of intelligence our society possesses. Computer networks, on-line databases, and satellite link-ups are all technologies that are being used to bring information to us on a daily basis. This information influx permits us to formulate decisions in a timely fashion. As we obtain and assimilate more data, our ability to make superior decisions increases, as well as our productivity. Yet, much of our time is still spent diligently searching for the right information.

Now envision 250,000 pages of information on a 120mm disc that can be accessed by a personal computer. CD-ROM is the product of that vision. The dilemma we are constantly struggling with is the transfer of knowledge (or lack thereof); part of the answer is CD-ROM.

What is CD-ROM?

CD-ROM stands for Compact Disc Read Only Memory. It is an optical disc having a diameter of 120mm made from the same material as bullet-proof glass. Information is stored on a CD-ROM in one continous spiral track that is approximately three miles long. The storage capacity of a CD-ROM is greater than 600 megabytes, which translates to 250,000 typed pages, or greater than two million sectors on a HP3000. The CD-ROM drive is connected to a personal computer, making it a neatly packaged solution. The cost factor for large volume pressing of CD-ROMs is also enticing. The latest quotes from mastering plants are anywhere from three to ten American dollars per disc. It should be noted, however, that the first disc, and the data preparation phase, is where most of the costs are incurred.

The CD-ROM disc cannot be written to, making it an ideal media for the collection and distribution of static information. It has a well-defined file structure, formerly labeled High Sierra, that has now become an international standard (ISO 9660). Codes used for the detection and correction of possible errors in the data are supplied with each block of data (2K bytes) that is written to the disc. These error codes are so efficient that only one error will escape uncorrected in 10 quadrillion bits (1 followed by 16 zeroes). Compact Discs have the capability for storing not only audio and text; but graphics, color photographs, full-motion video, animation, or any combination of the aforementioned.

Consider the Shortcomings...

How do we access all that data? There are certain unique characteristics about the hardware and the media that software developers must consider when outlining a strategy for the retrieval of information from a CD-ROM disc.

The first problem is in the physical speed of the drive. The CD-ROM drive rotates at a relatively slow speed when compared to most magnetic media of today, such as a Winchester hard disc. The drive operates at fluctuating speeds depending on where the laser is on the surface of the CD-ROM disc (faster on the inner tracks). This is known as constant linear velocity, or CLV. Most magnetic media of today operate at an unvarying speed or constant angular velocity, CAV. The CLV format is ideal for the storage of large amounts of data, but poor for the retrieval of individual blocks of data. The head movement of the laser must be accompanied by the mechanical process of speeding up or slowing down the disc. This movement accounts for high seek and latency times. Transfer rates of data coming off the disc fall between the rate of a high-speed floppy disc drive and a hard disc drive. The developer must consider these facts when creating and positioning files on the disc and strive to counterbalance the poor seek and latency times of the hardware.

One benefit of the CD-ROM could also pose a problem for some developers when dealing with more volatile types of information: the disc is read-only. Formulating an update scheme could involve writing the information provider's/user's input to another media (eg. hard disc) and displaying the most current information at retrieval time. Where the information comes from is totally transparent to the user. Response times will be visibly slower when multiple media-types are read. An alternative could involve distributing a new CD-ROM. The less static the information is, the more frequently new discs may have to be generated. Webster's Dictionary does not have to be updated as frequently as the market price of Hewlett-Packard stock!

Other Concerns

The types of data that the user will be accessing will play a major part in determining the design of your CD-ROM retrieval system. Numeric data will require certain options within the retrieval software that may not need to be implemented in an all text-based system. Graphics, audio, and video will all require certain elements be built into the interface for their retrieval and use.

Security can be a major concern. Due to the abundance of information that can be put on a disc, certain portions may not be accessible to some users. If that is the case, you may wish to encrypt the data on the disc as well as build stipulations for access into the retrieval software.

Graphics and video often require that special hardware boards be installed into the PC to decompress and display the information. A special monitor may be necessary as well. Can your user base fund this expense? If not, the feasibility of having data of that nature on your CD-ROM will be reduced.

The Current Alternatives

Upon deciding that CD-ROM is the answer to your needs, there are a few alternatives to compare in terms of how to construct the user interface.

You may wish to develop the entire interface on your own. This will take a great deal of time and money. Often, that "window of opportunity" of getting a needed product out to your users will pass by while the software is still in the development phase.

Another appoach would be to have an interface built for you by a company that specializes in custom software. This can be extremely expensive.

There are many companies specializing in CD-ROM that offer a full range of services to their customers. This covers everything from the preparation of the data, to a quality assurance check of the CD-ROMs that are returned from a mastering facility. They also offer a variety of user interfaces from which to choose. The intent is to pick one that fits your needs and the nature of your information. These "stock" interfaces can usually be customized to some degree. The issue with service organizations is that generally you receive something that does not quite meet your original expectations, although a product can be delivered within a relatively short period of time.

Finally, a few companies offer a set of software libraries that can be used for the development of a user interface. This concept is often referred to as a toolkit. The underlying search and retrieval software is provided, and the developer is given the ability to construct an interface on top of this basic "search engine". This provides the developer with full customization of the portion that the user accesses, without the total development time that would be required to build an entire system. Royalty fees are often required to license the company's software.

Features of Retrieval Software Packages

All CD-ROM retrieval packages offer the user and developer a wide genre of features from which to choose. This section will outline and briefly explain many of those features.

Keyword Search -	Most CD-ROM retrieval packages offer an inverted file management system. In this system, all significant words within textual data are indexed. The concept of the keyword search allows the user to type in a string of keywords, with full boolean constructs (AND's, OR's, NOT's and the use of parantheses), and retrieve a search result set. This set contains pointers to the data that indicate which documents were found that contain keywords and where within each document the keywords were found. A document can be described as the lowest "record" of data found within a CD-ROM database. This is generally one to five screens of textual data.
Expansion or "wildcard" -	This feature permits the user to retrieve all indexed terms that begin with a particular base. An example of this would be "COMPUT*", which would retrieve occurrences of such terms as "COMPUTER", "COMPUTATION", etc.
Phrase and Proximity -	Use of the proximity operator lets the user limit the area in which keywords are searched for. For example, [word1 word2] indicates that word1 must be found within a supplied distance of word2. This distance is configurable and can be specified in terms of characters or words. The phrase search is specified when a user wishes to look for an exact match on a series of terms within the indexed text.
Thesaurus -	Often a thesaurus can be invoked by the user that contains a list of synonym terms that have been specified by the information provider and/or the user. It may or may not be modifiable. By invoking the thesaurus, the user indicates to the search engine to retrieve all synonym terms for the supplied keywords and construct an "OR" condition using those terms. An example of this feature would be to have the user type in "DB", and via the thesaurus, the search engine would return a search result set containing references to the terms "DB", "DATA BASE", "DATABASES", etc.

Topic search -	Instead of searching for general keywords, pre-constructed search result sets may be accessed by the user which provide a quick pathway to information relating to a specific topic. The topic "COBOL SORT ROUTINE" may contain pointers to documents such as application notes, software status bulletins, and manuals that all contain information relative to that topic.
Range search -	This is a critical feature for designing an interface that will be accessing numeric data. This allows the user to specify a mathematical range when searching for specific data.
Field search -	The user may specify certain keywords that will be searched on only within a pre-defined field in the database. An example of such a feature would be "NAME=BILL", where the search engine would look in the pre-defined database field "NAME" for all occurrences of the term "BILL".
Subsearch or "sideways search" -	A subsearch is allowing the user to search across a previously retrieved search result set. This lets the user further define the search to work with a manageable set of documents. Searching on "FILE" may retrieve hundreds of documents. Performing a subsearch on that search result, using the keyword "ERROR", will limit the amount of documents that the user has to examine, making a more precise search result set. A "sideways search" involves using a term from a document that has been found via a previous search. Using our previous example, the user may be reading through a document relating to "FILE ERROR CODES", see a reference to "SECURITY", and perform a "sideways" search using "SECURITY" as the search term. A new search result set will be constructed for the user to examine.
Relevance -	There are some search engines that will rank your retrieved results based upon certain algorithms; such as the number of times specified keyword(s) occurred within each document. A weighting factor is assigned to each document, and the search result set is sorted according to this factor, prior to being displayed to the user.
Search history -	This feature permits the user to save a query string with all related information about the search (ie. number of documents retrieved, date, etc.). The user can employ the search log to re-construct a previous query into the database for fast retrieval to information.
Browsing -	Browsing allows the user to see the hierarchy of information within a database and follow a path to specific knowledge. Paths may be pre-defined by the information provider, established by the user, or a combination of both. Browsing can be presented in different ways. The discussion on hypermedia presented in this paper talks about some of those ways. Often times a "topic explosion" approach is taken. The user is presented with a set of themes (ie. magazine articles). By selecting a theme, the user may now be presented with the specific text relating to that theme (ie. the article itself).
Book marks -	Through this feature, the user may establish a pointer directly to information that is of specific interest. Use of this tool gives the user faster access when needing to reference particular information again.
Annotation -	This will let the user customize the information with certain facts that make the data more relevant. These personalized facts may be referenced upon viewing the annotated document via a keyword search or by other means.

The LaserROM Strategy

LaserROM was developed to aid customers in their use of existing Hewlett-Packard documentation. This tool uses CD-ROM technology as it's "delivery vehicle", taking advantage of the enormous storage capacity of the CD-ROM. The use of a MicroSoft (C) Windows-based graphical interface in conjunction with CD-ROM makes LaserROM an attractive, integrated, PC-based solution. Monthly updates provide customers with current information that is immediately at their disposal within the framework of the PC workstation environment. Full keyword indexing and hierarchically structured browsing supply the user with direct pathways to the primary feature of LaserROM: the information (reference manuals, product catalogs, known problem reports, etc.). These pathways give LaserROM an added value above the printed copy. Additional benefits are described in the following sections. LaserROM is an entirely new way to look at documentation.

Why Windows?

MicroSoft (C) Windows was chosen as the primary operating enviroment for LaserROM for a variety of reasons. Windows provides the user with a heuristic, easy-to-use interface. The Hewlett-Packard commitment to this environment has been conveyed through the introduction of New Wave. The PC industry as a whole is now committed with the announcement of the next operating system, Presentation Manager, which is being developed under Windows. Windows is fast becoming a standard amongst the users of personal computers.

The New Wave Environment

The LaserROM development team was able to work with the the New Wave Human Factors engineering group within Hewlett-Packard, to help make the LaserROM interface conform to guidelines that had been established for development within the New Wave and Windows environments. This effectively gives LaserROM the same "look and feel" as other Hewlett-Packard applications developed within this environment. This approach benefits customers, allowing them to spend more time using the product and less time in the training phase.

The New Wave environment also gives LaserROM an additional advantage above the alternatives we previously discussed. New Wave permits the user to retrieve valuable information without leaving their normal working environment. LaserROM becomes part of a larger, totally-integrated, object-oriented office environment and not just a stand-alone application. The New Wave architecture will be compatible within the DOS, OS/2, and UNIX environments.

In summation, four key advantages were derived through the use of New Wave and Windows in developing the LaserROM user interface:

1) LaserROM was developed under an industry standard.
2) LaserROM is heuristic and easy to use, with a common Hewlett-Packard New Wave user interface.
3) The information is readily and quickly accessible.
4) It allows the user to retrieve information within the framework of their normal operating environment from a well-integrated workstation.

The Features

A key strategic point in the development of LaserROM was to concentrate on the content of the information. When a user is given tools, such as keyword searching, which allow for direct access immediately to relevant data, errors within the data become much more visible. For that reason, an extensive quality check was done to the data that was placed on the CD-ROM. Another primary design strategy was to give the user additional capabilities only if they were easy to use, not to implement features based on a technology-driven focus. Additional features will be added once they are deemed necessary from a customer standpoint, provided they fit into the overall LaserROM development strategy.

The main feature of LaserROM is its keyword search capability. The utilization of full boolean constructs, and a thesaurus created by the information provider, supplies the user with the ability to "zoom in" on data of high points of interest. Proximity searching, phrase searching, and the use of an expansion operator are all supported by the software.

Another way to access information is through the LaserROM browse facility. Browsing allows the user to look at how the information is layered through a "structure explosion" capability. For example, the user selects a database and is then presented with a list of all possible manuals within that database. Selecting a manual then generates an "electronic table of contents", and so on. At any time, the user may reverse the process by examining a previous level.

Because approximately fifty percent of Hewlett-Packard manuals contain graphics, being able to display that data type became imperative. Activating an icon within the text provides the user with the alternative of viewing the figure only if deemed necessary. It also allows the user to scroll through a document on the screen quickly, without the burden of watching a graphic imbedded within the text slowly re-paint. The figure is displayed within its own window, giving the user the flexibility of moving or sizing the window. This window may be left on the screen for the user to refer back to, while reading different sections of the text.

A complete, thorough help system was also fully integrated into LaserROM. It gives the user a generic index from which to access topics discussed within, and a context sensitive mode that can be used to see specific text relating to menu items or individual windows with their associated commands.

The Help subsystem, a computer-based training package that comes standard with LaserROM, and the heuristic nature of Windows all contribute to the fact that LaserROM does not require use of a manual. As a matter of fact, one was not printed for those specific reasons. The best training for LaserROM is to use it.

The Future in CD-ROM Information Retrieval

Clearly, CD-ROM is a technology whose time has come. The addition of interactive media and graphical user interfaces has made information assimilation exciting, rather than mundane and tedious as it often has been. It is human nature to take more interest in the things we enjoy doing, and to ignore or prolong the tasks we despise.

Many new technologies have been introduced that try to enhance the nature or content of information, as well as bring to the user an element of excitement and intrigue when using them.

Hypermedia

The theory of hypertext and hypermedia goes back to the days of the quote found at the beginning of this paper. In 1945, Vannevar Bush synthesized the idea of a "memex" - a complete information "storeroom" at the disposal of each individual that could be accessed and manipulated within the framework of the user's working environment. Information was intertwined with cross-linkages and was stored on microfiche (later updated to being on a computer). Theodore Nelson (circa 1965) has been credited with coining the term hypertext, and along with Douglas Engelbart, have become founders of the ideas that lie behind the technology. The simple concept of hypermedia is to allow users to establish linkages between all data types so that these linkages form a natural "knowledge path" that may be followed. This perception embraces the idea of non-sequential reading vs. normal, sequential reading. Linkages are generally not carried with the data, but rather stored in a separate database. The information provider may also establish linkages within the data, allowing the user to follow many different paths, if so desired.

Hypermedia can be thought of as a tool that allows users to "teach" the computer the way they themselves think. Eventually, software may be smart enough to emulate the knowledge processing traits of the end user and apply these rules to new information when it is added into the system, thereby automatically establishing the necessary linkages.

As you can imagine, the problem of updating these linkages and incorporating hypermedia into a changing environment is a question that software developers will be (and have been) wrestling with for quite a while. Two such tools currently on the market that deal with this technology are Guide and Idex(C) by Owl International and Hypercard(C), recently introduced by Apple.

Multimedia

With the use of technolgies such as CD-I or DV-I (Compact Disc- or Digital Video-Interactive), full-motion video on a CD-ROM disc is now possible. You can imagine the uses in the fields of home entertainment and especially interactive training. Being able to look up and read the textual portion of how to fix a disc drive head crash, invoke a detailed graphic of the drive internals, view a full-motion video sequence of someone fixing the drive (freezing the video when necessary), and having each step narrated-all on your personal computer-is not a concept that falls only within the realms of science fiction anymore.

Conclusion

The world of CD-ROM is growing. Products are being introduced that give the user more power in accessing information then was ever deemed possible before. The door is open to both information providers and software developers to coordinate their efforts and utilize the tools I have outlined. Through their efforts lies not just the future of information distribution and retrieval, but the future of computing in our society as well.

Strategies for Re-packing Discs
and
Image Data Bases
by
Michael Hornsby

Hewlett-Packard
4501 Erskine Road
Cincinnati, Ohio 45242

How do you know when its appropriate to reload your system or re-organize your data bases? Unfortunately, the developers of MPE and Image did not provide a user friendly yellow and red alert message system for this purpose. The answer to this question is highly dependent on the size, number and type of applications resident on the system in question. The purpose of this presentation is to provide insight into the planning and execution of activities required to maintain an optimum performance level on your system, while utilizing available disc space as efficiently as possible.

Topics to be discussed in this presentation:

How to review your system for over configuration of directory and virtual memory disc space.

Planning for disc space growth requirements.

Archiving files and IMAGE data bases on a periodic basis to remove files that aren't needed for online access.

Disc compression and reloading strategies.

IMAGE master data set disc space requirements and performance tradeoff's.

IMAGE detail data set packing strategies.

Using facilities of Turbo IMAGE to reclaim disc space.

I. In the Beginning

Your HP3000 arrives and your Systems or Customer Engineer installs the operating system and whatever subsystem software products you have purchased on the system.

A. The Initial Configuration

The initial configuration of your system will involve using disc space for the system directory, usually 6,000 to 10,000 sectors, and virtual memory 25,000 to 35,000 sectors per drive. Will your system require more or less than this? Only time will tell. The amount of space configured for the system directory and virtual memory on ldev 1 can only be changed by reloading the system, so these parameters are almost always over configured. More realistic values can be set after the system has been in operation for six to twelve months. A sysdump $null will divulge the amount of space configured and available in the system directory. A run of tuner will display the amount of space configured and the high water mark of space consumed in virtual memory. Disc space can be reclaimed from non ldev 1 virtual memory by down sizing its configured value on a cool start.

B. The HP Initialed Accounting Structure

When the system is installed or updated, the installation process leaves behind accounts, groups and files that you do not need for everyday operation. These not only take up space but also are a security problem, as they are found on all HP3000's. The files in the HPPL@, TELESUP,and SUPPORT accounts and the files in the USL, CREATOR, and DOCUMENT groups of the SYS account can be stored off and purged. The installation process also leaves a group of environment files in the ENV group that are not needed if you do not have an HP2680 or HP2688 laser printer connected to your system. A last note, the install job creates many accounts and groups that may not contain any files. These can and should be deleted to tighten security on your system.

C. Planning for Growth

When the system is new and many millions of sectors of space are available, the last thing anyone is concerned about is planning for the day when you'll be out of disc space. This is the time to implement a monitoring and tracking process for growth. Remember, even if the number of users and applications on your system remain stable, you will eventually run out of disc space if no plan is put in place to manage it! This plan must contain a strategy for naming of accounts, groups and files. If all users logon as USER and all files are located in the PUB group of accounts, it will be very difficult to determine appropriate disc space management procedures.

II. The Mature System

Once a system has been in operation for twelve to twenty-four months the discs will be filled and it will be someone's task to account for this scarce resource. A common solution is to simply purchase another disc drive; however, this strategy is only a short term fix. What follows is a brief lesson in disc drive archaeology.

A. States of Disc Space

Disc space can exist in one of three domains: Configured, System, and Lost. Configured disc space consists of the file directory, Virtual memory, and Spooled input and output files. The System disc space consists of Mpe files and free or available space. And finally Lost disc space are those sectors which have been deallocated from the free space maps but aren't recorded as any of the aforementioned entities.

Configured disc space can be calculated in the following fashion:

1. Directory Size
   Sysdump $null ________

2. Virtual Memory
   Sysdump $null ________

3. Spooled Input
   Showin status ________

4. Spooled Output
   Showout status ________

Total Configured Sectors: ________

System Disc Space can be calculated in the following fashion:

1. Free Sectors
   Run FREE5.PUB.SYS __________

2. Total Allocated Sectors
   Report ?.@ __________

Total System Disc Sectors: __________

Lost Disc space can be calculated as follows:

1. Total Disc Sectors Available:

Drive Model		Formatted Sectors	
#HP7908's ____	*	64,750	__________
#HP7911's ____	*	109,824	__________
#HP7912's ____	*	256,256	__________
#HP7914's ____	*	516,096	__________
#HP7920's ____	*	195,600	__________
#HP7925's ____	*	469,440	__________
#HP7933's ____	*	1,579,916	__________
#HP7935's ____	*	1,579,916	__________
#HP7936's ____	*	1,200,984	__________
#HP7937's ____	*	2,233,752	__________
#HP7941's ____	*	92,928	__________
#HP7945's ____	*	216,832	__________
#HP7957's ____	*	319,095	__________
#HP7958's ____	*	510,552	__________

Total Available Sectors __________

2. Total of Above
   Configured Sectors __________
   System Sectors __________

   Total Known Sectors: __________

3. Total Lost Disc Space
   Subtract (2) from (1) __________

B. Wasted disc space

Aside from being Lost disc space can be wasted in many other ways. Poor blocking factors, Editor "K" files, HPWORD WPSPACEF files, NMLG log files, system logfiles, old source files, and above all mis-sized data bases to name a few. An effort should periodically be made to correct these problems, because not only do they waste disc space but they extend the length of backups, and reloads.

Other frequently overlooked options are the compression of source and program files, and variable length records. The contributed library contains utilities for compressing and unpressing files. If you must store source online thess are methods to minimize the impact.

C. Archival

So what can be done about these files that waste space and time backing them up? The solution is simple store them off to tape with a date= option of the STORE command! This little known parameter makes it possible to identify, store and delete files that haven't been accessed since a given date. I have seen this free up as much as a half million sectors on more than one HP3000!

The topic of mis-sized data bases will be discussed shortly, needless to say Image data sets should not be deleted in the above fashion.

III. Compaction Options

Once files have been archived, and the extent of lost disc space has been ascertained, we have several methods available to contend with the fragmentation of the disc space it's self. Disc free space becomes fragmented over a period of time determined by the volatility of builds and purges of files. How do I determine if a fragmentation problem exists? For each disc drive divide the largest free area into the total free area. If this ratio is 1 then all free space is in one chunk, however if this ratio is less than 20% then your largest chunk pales in comparison to the many smaller less useful chunks out on the drive.

A way to determine fragmentation is to issue the following command:

BUILD TAKSPACE;REC=128,1,F,BINARY;DISC=15000,1,1;DEV=1

If this command fails, your system does not have a contiguous chunk of 15,000 sectors available on LDEV 1. Not having 15,000 contiguous sectors on LDEV 1 will present serious problems if you attempt an update of your system. Therefore it is highly recommended that after an accounts reload you build the above file. By purging this file prior to an update, you will not have to subject yourself to the time and effort to create the space required. Purging this file also is a quick fix for the out of disc space all spool queues shut situation.

One other note about updates and LDEV 1. During an update the system SL is read in from tape and placed on the disc at the first contiguous area large enough to hold it. In the other load operation types the system SL is located next to the system directory, since both are highly accessed, this is an attempt to keep the disc head over the same location. Sever performance degradation can be experienced after an update if the system SL happens to be placed far away from the system directory. Disc caching has lessened the impact of this problem ,but it is a good practice to check the placement of the system Sl using listdir5. This could also explain why some sites experience different performance symptoms after updates to new versions of MPE.

To recompact your disc you have two dramatic options, the full reload, and the Vinit compress option, both of which will make your system unavailable for some period of time.

A. Types of Reloads

Reloading the system from a full back up is a time consuming process; however, it provides the best method of reinitializing disc space, in that when complete, the largest free area on each drive will be equal to the total free area on each drive. The reload accounts option is the most frequently used type of reload. This entails a quick reload to get the system up and running, then a restore to replace all of the user files. The compact option type of a reload can be used on smaller systems with two or three reel full sysdumps. In this type of reload, the system does not become available until the last file is restored from the last reel of tape. And if anything should go wrong, the only option is to start over from the beginning.

B. Vinit Compress

An alternative to reloading the system is the compress option of the Vinit Utility. This will compress a given drive's allocated files in an attempt to coagulate the free chunks. This procedure does not completely achieve the same state in a reload, in that the largest free area may or may not be increased.

C. File Copy Utilities

A free space increase can also be achieved by copying files to other disc drives. There are many utilities available for performing disc to disc copies. However, the same function can be achieved by storing off the desired files or files and then restoring them with a dev= option.

IV. Image Data Bases

On most transaction processing systems, the Image data bases consume the lions share of the available disc space. The following is a brief discussion of how to increase the performance and throughput of a system while recovering wasted disc space as an added bonus.

A. Master Data Sets

Probably the least understood and most maligned topic regarding Image data bases is the master data set, this is normally due to the fact that designers tend to put the cart before the horse and begin the design by specifying all of the possible keys into a detail or by defining stand alone manual masters that substitute for KSAM files. This of course leads to great performance and design problems it that only one unique key is available for direct access to the stand alone master, and all other searches require a serial search of the all the master records.

A different approach to design that leads to greater flexibility is to begin the design by specifying only stand alone detail data sets, and then choosing no more than three paths into any given detail. The utility of the fourth and greater paths usually can not justify the disc space and CPU cycles to maintain them.

1. Sizing Myths

The largest myth concerning master data sets concerns sizing, the thirty percent free space and prime number rules of thumb have had much discussion in the past. It can simply be demonstrated that for many masters these rules of thumb do not apply. Two examples come to mind. Short character keys ie dates tend to generate high percentages of secondaries at low fill levels, due to the sameness of content and low number of bytes to attempt to hash on. On the other end of the scale are masters with capacities in the half million or greater range, here different prime numbers or products of prime numbers can generate poor hashing profiles.

The only way to circumvent these problems is to use a utility to monitor the number and length of secondary chains. My personal preference is DBLOADNG, it not only reports on secondaries but also reports the number of average blocks that have to be read to access one master. Note, the perfect case is when no secondaries occur, and thus only one block need to be processed to access one master entry.

Thus in over sizing certain masters, especially ones that exhibit poor hashing performance, we can trade off disc space for CPU cycles.

2. Combining Automatic Masters

Probably the least understood image procedure is the dbfind intrinsic. This procedure is the heart of the power of image. It is used to set up for a chained read. The parameters of this call reference the detail data set and search item, and from this image knows which master to access. Thus two automatic masters with paths into a given detail can be combined if they have the same data type and size. A good example of this would be an order header detail with order date, ship date and invoice date as automatic masters. The same functionality could be gained with one date master with three paths into the same detail. This not only simplifies the database design but can result in rather substantial disc space savings. Note that this method can also be applied to different data items, for example customer number and invoice number, again if they have the same data type and size definitions.

Surprisingly enough, it turns out that due to the way the dbfind intrinsic works, this is a rather easy change to retroactively implement.

B. Detail Data Sets

As can be gathered from the above discussion I am highly disposed toward detail data sets. Experience has taught me that most Image performance problems can be traced to putting data in master's. The following discussion about detail data sets is the product of years of experience from the school of hard knocks.

1. Picking and Packing the Primary Path

The most important assumption that is made during the design of a data base is the length of an average chain! This value is intimately tied to the application that the data base is apart of, and should be well documented and reviewed during the life of the data base. For example, the number of line items per order, the number of orders per customer, the number of orders per day. If the answer is one, ie the number of customers per customer number, then this path can be said to be perfectly packed because, the number of IO's needed to read the entire chain is always one. However as the number of entries on a given chain rises in respect to the total number of entries in the data set, the usefulness of the path decreases. In fact if the average chain length grows past twenty percent of the number of entries in a detail, it would be more efficient to serially read the entire data set, as fewer blocks would be read.

So, if the average chain for a given path exceeds twenty percent of the detail data set, the utility of the path is highly suspect, and in all probability it could be eliminated.

To pack a given path, or set of chains we have several alternatives. One method is to unload, erase and reload the entire data base. This can be extremely time consuming. Another method is to use the DETPACK feature build into ADAGER, and the third method is to write your own custom detail packing utility.

2. Small Detail's Tied to Large Masters

Nothing wastes more disc space than tying a small detail to a large master! It is not at all unusual to find a hundred thousand entry master with a path into a detail with capacity for three or four hundred entries. Six words of chain head space is reserved in each master entry. If the detail is less than 300 entries, make it a stand alone detail, and simply serially search it. If the detail is larger, tie it to a separate smaller automatic master.

3. Unused and Unneeded Paths

After a system has been in production for many years it is very difficult to determine which paths are being used and which ones are a waste of disc space and CPU cycles. One method is to code traces into applications, this method while direct is difficult to implement and adds overhead to the application. Another way to get at the same result is to use the Profiler built into turbo image. The Profiler can be purchased, but it probably would be better if consulting were purchased from HP to interpret the results. Either way the performance improvements from reducing the number of paths into highly volatile details can be dramatic.

4. Unused and Unneeded Data Items

Not too long ago, the only method to modify image data bases was to use the dbunload, delete data base, change schema, build data base, dbload process. This achieved the desired end result, but not only did it consume great amounts of time, but filler data items where required to hold the place of changed or expanded data items. This resulted in a tremendous waste of disc space. These filler data items can easily be removed with the utilities now available.

C. Large Image Logfiles

Prior to turbo image, if your application wanted to implement disc transaction logging, you had to build extremely large logfiles, two or three times larger than ever would be necessary. This was due to the fact that if the logfile filled up, all processing against the data base would halt and the data base could become corrupted. Turbo image has the facility to log to successive numbered logfiles. Thus these logfiles can now be down sized by factors of two or three times.

V. Performance Implications and Conclusions

Review your system configuration of directory and virtual memory on a timely basis, overconfigured values waste disc space, underconfigured values require a reload to expand.

Understand the HP provided accounting structure and remove any accounts, groups and files you don't require, only after backing them up to tape.

Plan for growth of disc space requirements, even if the number of terminals on your system isn't. Running out of disc space can at worst cause a system failure, but at least will interrupt your production system.

Review the states of your disc space on a timely basis. Characterize the need to manage each category on a scheduled basis. Spooled print files tend to grow as the amount of data stored on the system does.

Purge the trash files off of your system on a periodic basis. This will make more valuable disc space available, and in the bargain speed up back ups and realods.

Archive files on a periodic basis to remove files that aren't accessed.

Use VINIT compress as a strategy to deflect the need to reload your system. Periodic compresses will coagulate available space, and thus the period of reloads can be lengthened.

Plan your reloads! It's better to do them on a planned basis than on an emergency one.

Be critical of manual master data sets, make them justify their existence.

Combine automatic masters of same data item type and size.

Track the average and maximum chain lengths in each detail data set. Watch for chains that exceed twenty percent of the detail capacity.

Pack the primary path of your most frequently accessed details. This activity alone can reduce response time by an order of magnitude.

Make small details tied to large masters stand alone, or give them a separate key.

Remove unused or inefficient masters, paths and data items.

If you have converted to turbo image, implement sequential logfile numbering, and down size your transaction logfiles.

These are only a few points to ponder. It is clear that disc space can be traded off for CPU cycles. The most frustrating SE situation is telling some distraught system manager that the only solution to the problem at hand is reloading his or her system, or worse data base. The only way I know to avoid these problems is to take the time to implement some or all of the above.

TITLE: Use of CD-ROM Technology for Information Services

AUTHOR: Phil Palmintere

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 2021

Remote Configuration Tracking: The CONFIRM Concept

Robert Poling

Product Support Division
Supportability Methods Lab
100 Mayfield Avenue
Mountain View, CA 94043

Introduction

CONFIRM (CONFiguration conFIRMation) is not a product. Nor is it the title of a project. CONFIRM is a support philosophy: to provide HP's support organization with *centralized* access to configuration information for remote customer systems.

Problem resolution has traditionally been viewed as a support service which applies to a system or group of systems belonging to a single vendor. When problem resolution required configuration information from the customer system, the support engineers had the following options. Depending on the level of sophistication required to get the data and the amount of data needed, they could request data from the customer. Large volumes of data would have to be transmitted or mailed to the support engineer. Alternatively, an on-site visit by a field engineer could be scheduled -- costly in both time and money. The increasing size and complexity of systems and the growing use of networks placed increasing importance on the availability of configuration information and required new support strategies for problem resolution.

Major developments in HP's support strategy include a focus on proactivity, remote support, and configurations. First of all, in support, *proactive* means dealing with problems by anticipating and preventing them, rather than reacting to problems after they have occurred. Second, HP's goal is to provide as much support on a remote basis as possible, in order to take advantage of centralized expertise of support personnel and to minimize travel time to a customer site. The emphasis on proactive diagnosis means monitoring events on a customer system and identifying problems before they occur. Third, there is a major focus on configuration since mis-configuration of hardware or software is a frequent source of problems, especially in networking environments. Configuration simplification and validation is an efficient method of avoiding or solving a large percentage of system and network problems.

CONFIRM is a natural extension of these support strategy developments which HP is pursuing by implementing tools for *remote configuration tracking*.

The goal here is to examine CONFIRM both as a conceptual development in support and as it is being implemented at HP. This paper will discuss in detail why remote configuration tracking is needed and what its benefits are. It will also describe the operation of existing remote configuration tracking tools and how CONFIRM fits into HP's support process.

Purpose

Remote configuration tracking has four major functions: retrieval, change detection, reporting, and centralization. The *retrieval* function performs the task of collecting configuration information from various sources on the customer system. The techniques used to extract and format configuration data depend on the source and may include low-level access to system data structures, database queries, or execution of another process to generate configuration data. The *change detection* function monitors the configuration for any changes which have occurred. Each change is noted as a configuration *event*. The *reporting* function handles the transfer of configuration events from the customer system to a central site via a data communications link, where the events are used to maintain up-to-date configuration information. And most importantly, the *centralization* function facilitates access to the data by support personnel. The user interface provides support engineers with a coherent, standardized way to access widely varied types of configuration data and multiple configurations.

Remote configuration tracking has evolved from the need for improved troubleshooting, timely access to configuration data, and access to multiple configurations.

Historically, configuration information has been collected from customer systems by remote support personnel on a *reactive* basis; that is, as needed. Interactive remote support has improved the situation by providing faster access to data and reducing the demands on customers in the support process. However, the effectiveness of interactive remote access is limited by slow data transfer rates, restricted system accessibility, and cumbersome user interfaces for data retrieval.

These difficulties are further exacerbated by the rapidly increasing size, speed, and complexity of computer systems and computer networks. Larger and more varied applications, distributed computing, multivendor networks and increasing user sophistication all contribute to the growing importance of configurations in support. As a result, problem resolution requires configuration information which must be retrieved from multiple sources on one system, must be current since configurations are increasingly dynamic, and must be available from multiple systems when data communications is involved.

Remote configuration tracking provides centralization for customer system configurations using an automated process. It gives support engineers fast access to formatted configuration data which is accessible regardless of customer system availability.

When configurations are tracked remotely, the customer benefits directly from increased system uptime. It takes HP less time to solve problems when they do occur, whether detected reactively or proactively. And, the customer is less burdened by involvement in the support process.

Operation

The discussion on operation of configuration tracking is in two sections. The first section is an overview of the design and data-flow of remote configuration tracking tools as they have been implemented at HP. The latter section is a survey of existing implementations which will provide more detail on the types of configuration data tracked by each tool.

Overview

HP's Predictive Support provides the support organization with a proactive tool for detecting potential problems before a system's operation is noticeably affected. It is provided as part of HP's Standard, Basic and Guaranteed Uptime support contracts. In addition to its proactive function, the A.02 release of Predictive Support includes several utilities which provide remote configuration tracking for the HP3000 family of computers. These utilities share a common functional architecture which can be divided into three components: onsite, datacomm, and Response Center. The *onsite* component performs the functions of retrieval and change detection. This component represents one of the Predictive tracking utilities. The *datacomm* and *Response Center* components perform the functions of reporting and centralization, respectively. These components are provided by existing Predictive Support software. An operational overview for a Predictive tracking utility follows a sequence of steps which is illustrated in Figure 1.

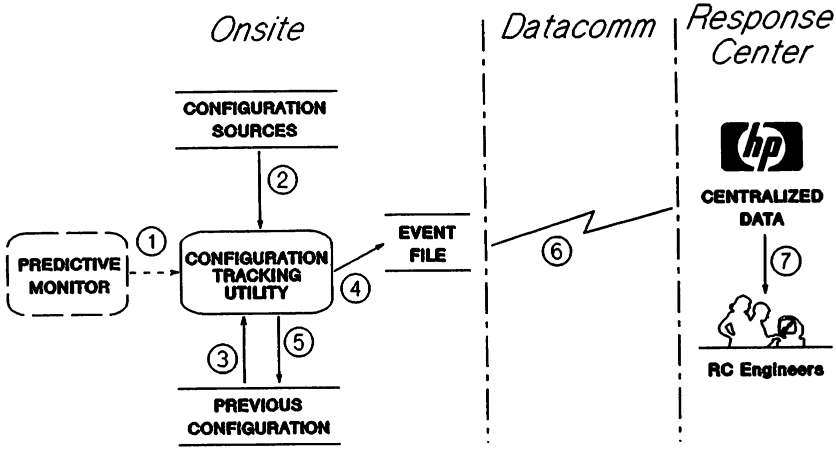


FIGURE 1 – Operational Overview of Predictive Tracking Utilities

1. Predictive A.02 is run every twenty-four hours on a customer's HP3000. Each time Predictive Support software runs on the customer system, it executes the tracking utilities as child processes.

2. The tracking utility collects data from the system sources which contain the current configuration information. If necessary, the raw data is parsed, type converted, and/or reformatted.

3. The current configuration is compared to the contents of the previous configuration file which was kept by the utility during its last run. Note that non-configuration information (current date, number of accessors, etc.) are masked or ignored to eliminate unnecessary configuration events.

4. Any differences in configuration are timestamped and reported to the event file. If the previous configuration file does not exist (as for a first run), it is created and the entire configuration is sent as events to the event file.

5. The previous configuration file is updated to reflect any changes in the configuration; new or modified entries are timestamped.

6. Software on both the customer system and the Response Center system is required for the datacomm component. The datacomm software on the customer system establishes a modem link to the Response Center System. Events generated by the onsite software are transferred across the datacomm link to the Response Center system. The datacomm software on the Response Center system verifies the integrity of the transferred data and stores it.

7. The stored data are then processed by Response Center software which uses the

events to update the configuration in a knowledge base. This component also provides the interface through which Response Center Engineers access the knowledge base.

These Predictive utilities all follow the same basic steps for centralization of configuration information. The major differences between them are the type of configuration information they monitor and the access methods they use to collect that information.

Existing Implementations

Three configuration tracking utilities are being released with Predictive A.02. These utilities monitor a customer's network configuration, system software configuration, and hardware configuration, and are called NETTRACK, SWTRACK, and SYSTRACK, respectively.

The network configuration utility, NETTRACK, gathers its information from a number of different sources and subsystems on the HP3000. Figure 2 illustrates how NETTRACK processes network configuration information and generates network configuration events.

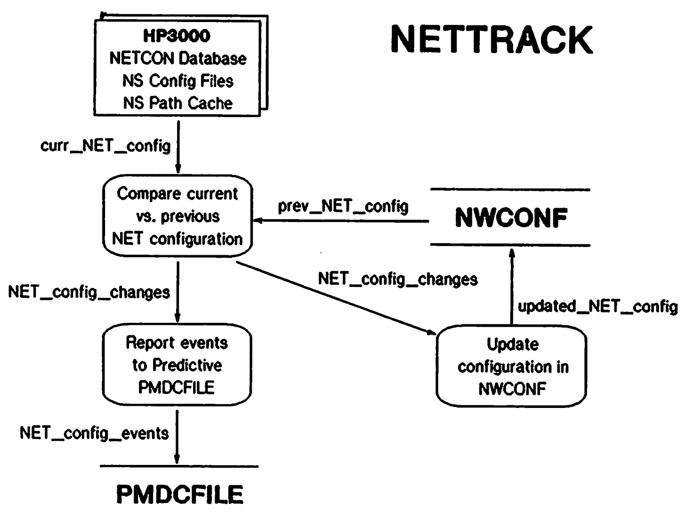


FIGURE 2 - NETTRACK Data Flow Diagram

When NETTRACK is activated it collects the current network configuration into a temporary file. It then compares the current configuration to the previous configuration in data file NWCONF for any differences. Any changes which are detected will be reported as configuration events to the Predictive event file, PMDCFILE. The temporary file then becomes the new NWCONF file. If NWCONF did not exist (as for a first run), the temporary file becomes NWCONF and all data are processed as configuration events. Predictive datacomm will handle the transfer of events to the Response Center.

This information is collected every time NETTRACK is run and will range in size from approximately 80 to 350 records. The number of events generated will depend on how dynamic the network configuration is.

NETTRACK collects and maintains the following information:

- DS X.25 Remote Node information, which includes node name, PDN address and phone number, and additional information obtained from the NETCON database.
- DS X.25 Line Characteristics, which includes logical device number, local PDN identifier, name and address, level 2 and level 3 parameters. This information is also obtained from the NETCON database.
- Network Directory information, which includes global and directory entry information from the Network Directory file NSDIR.
- LAN and Router remote node information, which are derived from NS path cache reports and maintained over time to provide an approximate remote node topology for these networks.
- NS/3000 Critical Summary information from resident NS configuration files (NMCONFIG, NSCONF, and DADCONF).

The second configuration utility, SWTRACK, keeps track of information for installed software program files, SL files, and SL procedures. The operation of SWTRACK is similar to that of the other utilities, with the addition of *maintenance* files. The CONFBILD maintenance file contains the list of software modules to be tracked. For program files and SL procedures, CONFBILD also specifies how and where to obtain associated software versions. For each SL file being tracked, SWTRACK generates a maintenance file which contains SL segment information.

The following information is monitored by SWTRACK:

- Program file information including filename, product number, version (vuf), actual and computed checksums.
- SL file names and segment information including segment name and length, checksum when segment was prepped into the SL, current checksum, and date of last change to entry.
- Name and version of SL procedures.

The third utility, SYSTRACK, covers hardware configuration. The main sources for this information are system memory, system configuration files (CONFDATA and DEVDATA), and the Predictive configuration file.

SYSTRACK maintains the following hardware data:

- System specific information including MPE system id, MPE release id, CPU type, memory size, basic table sizes, scheduling and spool information.

- Volume information for each system and private volume including logical device number, volume identification information, and virtual memory information.

- Device class information for each device class including class name, access type, and logical device numbers for each class.

- Logical device information for each ldev including ldev and DRT numbers, mode flags, class information, and other device configuration values.

- CS information for the communications system and CS details for each CS device including ldev number, port mask, protocol, modes, timeouts, speed, buffer sizes, and driver options.

Since these utilities run within the framework of Predictive Support, customer security is protected by existing Predictive security.

Strategy

Consider then, how remote configuration tracking fits into HP's overall support process.

When a potential problem is detected proactively by Predictive software, it is automatically reported to the Response Center; a problem being handled reactively will be manually called into the Response Center by the customer. Once the problem is classified, a Response Center Engineer (RCE) will determine what, if any, configuration information is needed to aid in problem analysis.

For example, a problem is detected in the X.25 communications between two systems. First the network configuration is checked for compatible addressing, window sizes, etc. Depending on this comparison and other reported symptoms, the RCE may then look up the software configurations of the two systems to check for incompatible datacomm software versions, or the RCE may examine the hardware configurations to verify that the INPs (Intelligent Network Processors) being used for the data link are configured correctly.

In the past, the RCE collecting all of this data would have been required to use 1200 baud remote data links to interactively access the customer systems. Assuming that the RCE could access the customer systems, it would take 30 minutes to an hour per system. But now, the RCE can access multiple configurations through a single interface, at direct connect speeds.

In another scenario, a customer reports intermittent data transfer errors on a LAN which have been occurring for the last week. Before resorting to exhaustive datacomm debugging techniques, the RCE could scan the configurations of systems on the LAN to check for any notable configuration updates which are timestamped as of one week ago. This data could be used to focus problem resolution.

These examples are somewhat simplistic, but they illustrate the usage of configuration information during problem resolution and the usefulness of remote configuration tracking in the support process.

The CONFIRM concept places configuration data at the support engineer's fingertips when it is needed, in a format that is easy to use. It is not a new support strategy, but a refinement and enhancement of HP's existing approach which integrates hardware and software support with a full range of support services, from consulting services to problem isolation on multivendor networks.

Future implementations of CONFIRM can be targeted for additional systems: dedicated hardware devices as well as general purpose computers. The localization of configuration information will also facilitate development of tools to automate configuration validation and comparison. Using artificial intelligence techniques, tools can be developed to proactively detect potential configuration problems and possible solutions.

Conclusion

HP is dedicated to helping customers maximize their system availability and performance while minimizing their operating and maintenance costs. The remote configuration tracking tools presented in this paper achieve these goals by centralizing customer configuration information where knowledgeable support personnel can access and analyze it, providing timely response to customer support needs.

The CONFIRM concept is a foundation for future directions in support technology and an integral part of HP's support strategy.

Managing MPE For Support

Bill Sutton
Hewlett-Packard
North American Response Center
Atlanta, Georgia

I. Introduction

Imagine the scene -- a desperate system manager at 2:00 in the morning. There is smoke pouring out of his Series 70 and the bit bucket threatens to catch fire. Our intrepid hero calls the Response Center for help. After ten minutes of frantic nail-biting, he receives the call. As the soothing tones waft across the WATS line, what does our brave system manager hear?

Requests for information.

LOTS of information.

What is the problem (in detail)? What subsystems are involved? What are their versions? What version of MPE are you running? When was the last backup? When was the last UPDATE? When was the last COLDSTART? When was the last hardware change? When was the last software change? When was the last configuration change? Has this mix of applications ever worked before? Who was on the system and what were they running? What...when...how many...how long...<puff, puff>.

The questions can be frustrating at times, but all of them are necessary. To determine the cause of a problem, support personnel (whether hardware, software, or business) need to know the environment of the problem. There are many cases (ask other system managers at SIGBar and you'll hear lots of them) where the solution to the problem was in an obscure, cobwebby corner that never could have been searched without a complete set of answers to the endless questions above.

We can't get rid of the questions (of course, someday HP may develop an ESP interface option for PICS, but that's a future and we're not allowed to talk about it). The best we can do is to speed the process of getting those questions answered.

How can we speed that process? Unfortunately, until that ESP device gets to release 2 (Read-System-Mind-MIT), there is little we can do from the Response Center -- keeping a local database of customer configurations lags actual customer changes, dialling in can take as long as getting answers to the questions on the phone.

There is good news, however. There are techniques that can be used by the venerable and wise system manager to keep the information most often needed both up-to-date and readily accessible. Of course, first she needs to know what information to have and how to get it. Then comes the arcane rituals of job streams, gold book updates, low mumbles of the magic words "vuf" and "vufi," and all manner of trivia which up to now was passed down only in secret circles from grizzled system manager to wide-eyed innocent young assistant system manager.

No more! For at risk to my own life from the SPASMP (Society for the Preservation of Arcane System Management Procedures) I will reveal the forbidden secrets of:

Getting Fundamental Operating System information,
Getting HP Product information,
Getting Patch information,
Getting Error information,
Keeping System Activity information,
Keeping Application information,
Locating Important Tapes and Configurations,
and Tracking Backups and Coldloads.

II. Definitions

Before things start humming, it would be nice to make sure we are all talking about the same things when we talk about Support, System Management, and MPE. All of the above words or phrases are subjective; they have a slightly different meaning to each person who uses them.

Support is the most broadly-based term used here. Taken to extremes, every activity geared toward keeping a system up and running can be considered support -- including the people who run the vacuum cleaner in the operators' lounge. This is obviously too extreme for our purposes, unless we want to be here until August 1990. For use in this paper, Support will be defined as activity which keeps the hardware and Fundamental Operating System software functioning so that applications may be run.

System management is another sticky term. There are basically two types of system managers -- the administrative manager (in charge of purchasing and budgets) and the technical manager (in charge of bits and bytes). Sometimes both types are rolled into one. In our case, the System Management activities we are discussing are those performed by the system manager who spends his days with his sleeves rolled up and his fingers poised on the keyboard. In other words, these techniques are used to help discuss technical problems with technical support personnel.

Finally we get to MPE. Confusion reigns here, as we have MPE V (the old new operating system), MPE XL (the new new operating system), and MPE V/R (the new old operating system). Suffice it to say that unless otherwise stated, the procedures here refer to MPE V systems and should also apply with minor changes to MPE V/R systems. There will be MPE XL examples scattered through this paper as needed.

All clear? Let's continue on to...

III. Vital Information

Vital information covers all those things you will almost always be asked to provide (system model, operating system version, etc.) Each piece of information is treated separately, with a few words on what it is, how to get it, what it is used for, and how to keep it up to date.

FOS VUF: Sounds like a French pastry, doesn't it? Actually, this is the Version Update Fix of the Fundamental Operating System. The Fundamental Operating System is all the software that comes with the system from HP at no extra charge -- this includes TurboImage, KSAM, Sort/Merge, the command interpreter, the file system, and many other modules. Version Update Fix is the version of the software and/or the operating system -- so called because it is in the form V.UU.FF (where Version = G, Update = A2, Fix = 04 {G.A2.04} for instance). The main VUF for the operating system is found on the SHOWME command:

```
USER: #S1751,BILL.SUTTON,INTEREX        (IN PROGRAM)
MPE VERSION: HP32033G.C2.02.  (BASE G.C2.02).
CURRENT: THU, MAY 12, 1988,  8:54 AM
LOGON:   THU, MAY 12, 1988,  8:36 AM
PROGRAM'S CPU SECS: 10        CONNECT MINUTES: 19
$STDIN LDEV: 24          $STDLIST LDEV: 24
```

The FOS VUF is the version you will be asked for most often. KNOW IT WELL. Often, a particular FOS VUF will have a name associated with it (G.02.00 is U-MIT, for instance). Always use the exact version number from the SHOWME command.

The scheme used for setting up the FOS VUF seems a bit strange until you get used to it. As a special added bonus, here is a quick rundown on each section.

The V (version) is the BASE RELEASE identifier. Versions of the OS with the same first letter will be based on the same tables layout, internal structures, etc. A change in the first letter signifies a major change in the OS direction. The most recent example would be from MPE IV (C.xx.xx) to MPE V/P (E.xx.xx) to MPE V (G.xx.xx).

The U (update) is the MIT identifier. A change in this number/letter combination signifies changes to particular modules of the OS which may or may not affect the entire OS. An example of this would be the change from T-MIT (G.01.xx) to U-MIT (G.02.xx).

The F (fix) is the DELTA identifier. A change in this number signifies patches integrated into the release, new products, and/or new versions of old products.

What makes all this confusing is the fact that, in the past, patched versions and special versions of operating systems have been indicated by adding letters to the VUF (for instance, Q-Delta-2 was C.01.02 but the Q-Delta-2 Product Tape was C.B1.A2). Rest assured that this won't happen again until the next time it happens.

For MPE XL, things will be different. Each position in the VUF will represent a particular set of changes and the numbers will be sequential (0-9,A-Z). We might visualize an MPE XL VUF as M.CR.PN, where:

N: NL BUILD or POST-MR BUILD. This is a new creation of the Native Library either for special customer requirements or to integrate last-minute patches after the software has been released.

P: PATCH BUILD. This is the normal 'fixed in the next release' next release. It does not include new products or enhancements.

R: PRODUCTION RELEASE. This is a release which introduces new products or enhancements to current products. This type of release would also add support for new peripherals and software.

C: CORE RELEASE. This release contains major changes to the OS, TurboIMAGE, or datacomm. A core release will increase the functionality of the product changed.

M: MAJOR RELEASE. All bets are off, we rewrote a whole bunch. This position will always be a letter.

PRODUCT VUF: When your question or problem involves a specific product, it is the version of the product itself that matters. All HP software products (in other words, software that has a separate product number than FOS) have separate version numbers. Most third party software will also have a version associated with that package and/or individual modules as well. For HP's products there may be program versions, SL routine versions, XL routine versions, and overall versions.

For most products, the best way to get the version number is to run the program and look at the banner. It's usually a little tough to run an SL entry, so HP products with SL version numbers provide an interface you can use to get those numbers. Let's look at TurboIMAGE as an example:

```
:run query.pub.sys
```

```
HP32216C.00.05 QUERY/3000  THU, MAY 12, 1988,  8:57 AM
COPYRIGHT HEWLETT-PACKARD CO. 1976

>vers

    QUERY     C.00.05

IMAGE PROCEDURES:
    DBOPEN      C.00.30
    DBINFO      C.00.26
    DBCLOSE     C.00.28
    DBFIND      C.00.15
    DBGET       C.00.29
    DBUPDATE    C.00.15
    DBPUT       C.00.20
    DBDELETE    C.00.20
    DBLOCK      C.00.27
    DBUNLOCK    C.00.27
    BIMAGE      C.00.27
TURBOIMAGE PROGRAM FILES:
    DBSCHEMA.PUB.SYS   C.00.28
    DBSTORE.PUB.SYS    C.00.24
    DBRESTOR.PUB.SYS   C.00.00
    DBUNLOAD.PUB.SYS   C.00.00
    DBLOAD.PUB.SYS     C.00.00
    DBUTIL.PUB.SYS     C.00.17

>e

END OF PROGRAM
```

If your product versions are different in any way from those shipped on the SUBSYS tape, we will want to know! In almost all cases, a patch to an HP product or SL segment will change the version number.

For you datacomm folks, you may see a slightly different version format. In addition to Version Update Fix, you will also see an Internal Software Level (I) listed for each individual part of a module. This level is NOT compared with other parts of the module to determine module compatibility -- in other words, we do not put out an error or warning if the Internal Software Levels do not match. We do put out an error or warning if the Version, Update, or Fix levels do not match. For instance:

```
run nmmaint.pub.sys

NMS Maintenance Utility  32098-20010 A.01.03  ...

Subsystem version ID's:
```

```
Node Management Services 32098-20010 module versions:

SL procedure:   NMVERS00                      Version:  A0103023
SL procedure:   NMVERSCSL                     Version:  A0103024
SL procedure:   NMVERS01                      Version:  A0103000
SL procedure:   NMLOGSLVERS                   Version:  A0103010
SL procedure:   NMLOGDATAVERS                 Version:  A0103014
SL procedure:   NMVERS04                      Version:  A0103007
SL procedure:   NMVERS05                      Version:  A0103000
SL procedure:   BFMVERS                       Version:  A0103002
Program file:   NMMAINT.PUB.SYS               Version:  A0103005
Program file:   NMFILE.PUB.SYS                Version:  A0103006
Program file:   NMLOGMON.PUB.SYS              Version:  A0103016
Program file:   NMDUMP.PUB.SYS                Version:  A0103043
Catalog file:   NMCAT.PUB.SYS                 Version:  A0103004

Node Management Services 32098-20010 overall version = A.01.03
                              :
                              :
                              :
                              :
```

As you can see above, none of the Internal Software levels match for NMS. However, since the VUUFF portion matches, the overall VUF is consistant and therefore the software is consistent.

Appendix A contains a partial list of HP software products and how to get their version numbers.

HARDWARE: For hardware-oriented calls (as in the smoking system example given above), the most important piece of information you have is the serial number. A serial number is required whenever the Response Center has to place a call to your local CEO.

Often, contract information is also useful. If you are familiar with your support contract (hours, response times, etc.) it will help you make a more informed decision about whether or not to call the CE out at 3:00 AM for a down printer.

To combine the two, make a copy of your hardware support contract. Write the ldev number next to the serial number of each device listed on your contract. Keep this copy in your gold book.

PATCH INFORMATION: Suppose we have a segment (or a SOM for you XL fans) that doesn't have a version associated with it. How do we find out if a change has been made?

Each segment has a unique checksum associated with it (quite by accident in the beginning). Nowadays we store the checksum as it was at compile time in the patch area of each code segment. There is a program in telesup called CHECKSUM (amazing how they came up with that name) that will extract the segment checksum and compare it to one that the program calculates. There are three reasons for using this program:

1) IF the checksums match (stored and calculated) we have a compiled version of the segment. In other words, nobody went in and mucked about with SLPATCH and changed the segment (of course, they could have changed the checksum once they changed the segment but it's doubtful).

2) IF the checksums do not match, we either had a binary patch applied or something trashed the segment. Good time to do an update.

3) We can compare the checksum as given by CHECKSUM to the checksum of a patch for that segment. If they match, the patch has already been installed.

Another way to tell (not as accurate but easier to read) is to look in the MPEMIT33/HPSWINFO file. All install jobs for patches from IND and CSY now include a section which places the date/time/patchid in the MPEMIT33 or HPSWINFO file. This info is at the end of the file.

Why is this not as accurate? It happens when the install job is streamed, not when the patch is installed. If the install job completes successfully but nobody ever does a COLDSTART, the patch really hasn't been put in but the MPEMIT33/HPSWINFO file will reflect that it has. Take the information in these files as an indication but not as gospel truth.

What about MPE XL? Currently we have two kinds of patches available -- SOM (Software Object Module) replacement and binary. Currently the most prevalent are binary patches, as the NL is all one HUGE SOM and therefore is difficult to replace.

Binary patch tracking on XL is done through the patch program (SOMPATCH) itself. Before any information in the SOM can be changed, SOMPATCH requires the user to enter logging information. All changes are then logged (old values and new values) and may be listed out as follows:

```
(SOM 0),ID: 523563  Duane Souder  SR: 4700-523563
                                  Wed Apr 13 07:11:58 1988
; ----523563----Compiler Library----Duane Souder
; U_resume_execution calls P_close_files on
; ICS and page faulted. 10/30/87 KF. Apply to
; 9.91, 9.92, X.01. To START image only.
;
```

```
; Offset changed on the X.01 line. Change offset
; from 7
U_get_escapecode          + 54    e85f1f8d | 8000240
```

Since this information is part of the SOM, if it is present then the patch is present.

Unfortunately, there is no equivalent to CHECKSUM on the MPE XL machines at this time. For the most part it is unnecessary, as all modules of the OS and compiler librarys have associated version numbers. It has always been a useful tool for users to use in tracking versions, however, and perhaps someday a checksum program for XL will show up.

ERROR INFO: All too often, diagnosis of a problem is impossible because the error that occured is (or becomes) unknown. There are cases in which errors speed by at a rate too fast for the human eye to capture, but there are also cases in which the user simply fails to note the error message. A knowledge of how to collect 'lost' error messages can help a system manager solve many problems without a call to the response center.

In the system environment, error messages may come up on the user's screen or on the console. If the message comes up on the console and gets lost (scrolls off, console gets cleared, whatever) CONSOLE LOGGING can save the day. Logging all console messages will vastly increase the size of your system log files, but looking at the log (entry type 15 for MPE V, entry type 115 for MPE XL) will retrieve information you sometimes didn't know the system told you. As an aside, console log records are also useful for security monitoring, tape drive usage monitoring (how many tape requests did you get?), and terminal usage (what ldevs logged on between 8:00 and 5:00?). To keep disc space usage down, make it a standard practice to store/purge log files immediately after each full backup. You might want to leave a few days' worth on the system for easy accessability.

Messages that appear on the user's screen and then vanish are much more difficult to find. Prevention and/or duplication is a much better bet. Here are some methods for tracking, finding, and displaying hidden error messages:

1) Perform all error checking possible within your application. When you do this, you can store the error number and message in a disc file or send it to the console and so avoid losing error info. In many languages (COBOL, for instance), if you do not handle the errors explicitly the program will abort however the system wants.

2) If a VPLUS screen is causing problems identifying an error message, redirect the screen to another terminal. You can do this by issuing a file equation on the filename in the VOPENTERM intrinsic call (example: FILE A262X,NEW; DEV=ldev; ACC=INOUT).

Now any non-VPLUS IO will go to the terminal that the program is running on, while VPLUS IO will go to the other terminal. Note: both terminals must be from the same product family when you do this (as in 262x, 264x, etc.).

3) Some HP Products log their own errors (TDP FINAL, NS, etc.). Errors on these products can be found after the fact by looking in the product-specific log file. Some products must have logging enabled for this to work -- refer to the product manuals for details.

4) On MPE XL, the CI VARIABLES may be used to pass error numbers and messages back to the CI (to a UDC or command file, for instance).

5) ESC 0 on terminals with printers hooked up can log all pertinant information if the users are trained to use it by habit. LOG BOTTOM or COPY ALL can be used for non-VPLUS applications. Set a standard that users cannot report errors unless they have a hard copy of the screen with as much error information as possible.

6) Also on MPE XL, errors from MPE subsystems are logged to a PROCESS ERROR STACK. Currently, not all subsystems use this stack (which is available in DEBUG with the pm_errors macro). However, it is still a good resource for tracking chain-reaction type errors.

TIPS: Here are some tips that could be useful to help you communicate vital information quickly and clearly:

1) Track the version number, not the MIT name or "The version of QUERY that came out on UB-delta-4." Using the exact version number of the OS and products saves you lookup time when checking for known problems.

2) The file HPSWINFO contains the released version of every HP product on the current release. If a product has problems, check to see if its version matches that listed in HPSWINFO. For those on earlier releases of MPE (pre-UB-delta-4 [G.B2.04]), MPEMIT33 serves the same function. If the version number is LOWER than that in HPSWINFO or MPEMIT33, you may have a bad version.

3) When you are asked for product version information do NOT use the HPSWINFO version number! You may have had patches installed, be running an old version due to a problem with AUTOINSTALL, or have fallen victim to any number of quirks which may cause the version to be different than that released with the system.

4) Keep the versions of frequently-used products close at hand so that you won't have to run the products. See the job streams

provided with this paper for examples of various methods used to track versions at coldstart/update time.

5) For MPE XL users, remember that the Command Interpreter and the Operating System have different versions now. What you see when the CI starts up is the version number of the CI, NOT the OS.

6) Information on patches applied since OS or product installation is good to keep as well. Keep the patch ID number, who got you the patch (RC, SE, by mail from SDC, etc.), and when it was actually 'coldloaded' into the system.

III. HELPFUL INFORMATION

The information mentioned above is usually sufficient to solve most common problems. There are times, however, when a problem is so nasty and vicious that it continues to hide deep in the bowels of the system despite all efforts to dig it out. This means additional, environmental information may be needed.

Everyone has run into a problem that only comes up sometimes, even in the same function of the same program. The solution to this kind of error almost always involves the interaction of the program with other programs running on the system at the same time.

How can we find out what these environmental factors are? Can we get general system trend information which will give us an idea of what might be running at a given time of the day? Here are some methods to get that very information:

SYSTEM ACTIVITY: Some disc errors, memory pressure-related errors, and OS table-related errors occur only at certain levels of certain types of system activity. If you know the way your system tends to run at certain times of the day, investigation of the causes of these problems becomes much easier (and faster).

Of course, the easiest way of tracking overall levels of system activity is to let us do it for you. HPTREND gives hourly averages of CPU usage, IO activity, disc usage, and other resource activity. This type of indication will help define if an intermittent problem occurs only during periods of high <fill-in-the-blank> access.

Suppose you either don't have HPTREND (why not?) or your usage changes daily such that a monthly average doesn't help much. There are methods which, though they take a lot of time and manual calculation, can give the same figures.

The tools are OPT, SHOWJOB, the streams facility, a spare terminal, and (optionally) a spreadsheet program. The general method is as follows:

1) Get the following data from the batch report in OPT on an hourly (or more often) basis.

   -CPU : Busy, Pause Disc & Swap, MAM-ICS, Cache MAM-PROC STK, Cache MAM-ICS, Overhead.

   -Disc: All I/O, Reads, Writes, Control Ops.

   -Launches: Process Launches, Process Swap-Ins, Process Pre-empts.

2) Take this information and plug it into your spreadsheet on an hourly (or more often) schedule.

3) If you can plot from your spreadsheet, you can directly see how busy your system is in given time intervals. If not, you may wish to graph the results over time.

For a detailed look at how various programs or users affect your statistics, you may add the following processes to your setup:

1) Set the interval for the job above to no more than 5 minutes.

2) Stream a job which does a SHOWJOB with output appended to a permanent file, then streams itself 4.5 to 5 minutes later.

3) Run OPT from an unused terminal and get a program report logged every 5 minutes (300 seconds).

The combination of all this information tells you which users and/or which programs tend to have the greatest impact on the activity levels of your system. They also tell you what programs are run at the same time under normal circumstances -- meaning that if an intermittent problem shows up it will be easier to spot any deviation from normal processing.

This is excellent information to have at hand, but the amount of time involved in manually transferring and calculating the results of this activity is very high. It may be worth it in shops where usage fluctuates wildly from hour to hour or day to day, and of course the technique is very useful once an intermittent problem rears its ugly head.

Unfortunately, supported tools to provide all of these functions do not exist yet on MPE XL. HPTREND should be available Real Soon Now, and other performance related products should follow.

APPLICATION INFORMATION: Just as a large number of calls are solved using standard information as listed above, a large number of calls are solved by tracing the onset of the problem to a change in application code. How do we find out whether changes were made, and how do we track who made them?

Unfortunately, HP does not rigidly force application version tracking. We do not require a version number as part of the object code, and we do not force an update to this version number with every compile. All of the methods available can best be expressed as voluntary and circumventable. For MPE V, we are very limited in our ability to imbed version information in object code. The most sophisticated method uses the Pascal $COPYRIGHT compiler option (also available in FORTRAN 77). Even though this places user written version information in the segment itself, there is no good way to print it out or access it.

One method of making sure changes have not been made is to always use the OLDDATE option on any RESTORE. This insures that the previous file create date (the compile date, for object code) is kept across restores. If a file's create date is recent, then, it follows that it is a new version of that particular program file.

In COBOLII, the DATE-COMPILED paragraph will bind a compile date into your object code. The WHEN-COMPILED reserved word may be used in your program code to reference or display this date.

On MPE XL, this issue is made a little easier. The program VERSION, which is in PUB.SYS and replaces PROGINFO and other unsupported utilities, accesses a part of the SOM known as the version area. In FORTRAN 77 and Pascal, a compiler directive, $VERSION, places user specified information in this area for VERSION to access. As an aside, the VERSION program also returns program capabilities, skeleton stack information, and maximum heap size, among others.

IV. TRACKING OS TAPES AND LISTINGS

If you have called the Response Center with a particularly solitary, nasty, brutish, and short problem there is a good chance that the solution path will come down to two options. One is to continue having the problem while a full-scale investigation continues. The other is to clean up the current version of MPE loaded on the system through an UPDATE or COLDSTART -- an action which will often clear the problem without a trace.

On a production system, clearing the problem is usually the choice. After all, the point of the system is to serve the users, not perform as a troubleshooting system for an obscure, one-time problem.

Too many times this solution fails because the caller could not locate a COLDLOAD TAPE from before the problem started occurring! In a worse scenario, the system is down, the only way to start it is from tape, and there is no tape to be found.

Anyone who has been in this situation has learned the hard way how important it is to keep track of OS tapes and their listings. It is equally important to toss old tapes and listings so that confusions (and sometimes career affecting actions) do not occur.

COLDLOAD TAPES: A COLDLOAD TAPE is defined as 'a tape created by SYSDUMP which includes system programs, drivers, the operating system, and @.PUB.SYS.' It is probably the most important tape you can have in regards to operating system integrity.

When you install your OS with AUTOINSTALL, the last thing it does is create a tape to load your system. THE INSTALL HAS NOT COMPLETED UNTIL YOU DO THIS! This tape is your ORIGINAL COLDLOAD TAPE.

From this point on, any time you make a configuration change, install a patch, or do any activity that requires creating a coldload tape, you will UPDATE from the IMMEDIATELY PREVIOUS COLDLOAD TAPE. This will insure that you have a completely clean version of the OS before you cut your new tape.

For example: you installed version X.YY.ZZ of MPE V last week. Now you need to add a printer. Before going into SYSDUMP to make the configuration change and cut the new coldload tape, you should UPDATE from the OCT (Original Coldload Tape). The new tape created by SYSDUMP will then have the exact same things (except for the new configuration) as the OCT, and any corruption which might have crept in during the week will not be propagated onto the new tape. This new tape now becomes the IPCT (Immediately Previous Coldload Tape).

From this the question arises: how many of these tapes should I keep? The answer, as usual, is that it depends on your site. If you do not make many configuration changes or install many patches, it certainly wouldn't hurt to keep all your coldload tapes for your current version of the OS. This gives you the option of backing up to any previous level of patches and/or configuration with just a single coldload.

If you have an active shop, however, the number of tapes could well be more than the amount of tape storage space you have. Your best bet would be to keep the OCT, the coldload tape just previous to the most recent change, and (of course) the coldload tape used to make the most recent change. You may want to keep more tapes than this if you change your system often (the more frequently changes are made, the more tapes back you may need to go to back out a problem). You should also store all patch files given to you before purging them from your system -- this way if you have to back out further than the number of tapes you kept you can still recreate your patch environment.

On MPE XL, the equivalent to the coldload tape is the SLT/STORE tape combination. Unlike MPE V, MPE XL does not store OS information and user files on the same tape set.

It is HIGHLY recommended that MPE XL users do a SYSGEN TAPE generation every time they do a full backup of the system. Also, for the store tape to be a true equivalent of the MPE V coldload user file portion, the store must be done with the DIRECTORY option.

BACKUPS: It only takes one RELOAD without a recent backup to discover that full and partial backups are necessary for the mental health of any good system manager. This is an example of Elephant Learning -- once you learn the hard way, you never forget.

Your operators should know where your most recent backups are. You should always keep at least the last full backup and all partial backups since on site. This is irrespective of security considerations, which may dictate restricted access to backup tapes and/or that all backups be kept offsite. Bear in mind that many system problems occur at night and gaining access to these tapes quickly could make the difference between a completed production run (albeit slightly late) and an open system management job. Seriously, security arrangements for tapes should include fast access for those authorized, not just safety for the tapes and privacy of the data.

OTHER CONFIGURATIONS: Many other products (datacomm, office products, etc.) require their own separate, more detailed configurations. Never forget that these configurations can be destroyed! Some sites maintain a separate tape which contains nothing but configuration files for their products (they use the indirect file feature of the STORE command to help in this). Keep in mind that often these products must NOT be in use (or even up) for the configuration file to be accessable.

V. IN CONCLUSION

The suggestions made in this paper are only a few of the ways to insure that important information about your system and its environment is always available. Every site has its own methods and standards for handling system information, but these methods should always include immediate access when needed as one of their goals.

If you follow the guidelines given above, you will find that the most tedious part of supporting your system -- gathering data -- will take less time and less effort. In addition, the data you have (especially in a down system environment) will be more accurate and therefore more useful to you, to HP, and ultimately to your users.

APPENDIX A
LIST OF PRODUCTS AND HOW TO GET THEIR VERSIONS

PRODUCT NAME	METHOD (MPE V)	METHOD (MPE XL)
ADCC SOFTWARE	:RUN TERMDSM.PUB.SYS	
ATP SOFTWARE	:RUN TERMDSM.PUB.SYS	
BASIC	:BASIC	
BASIC COMPILER	:BASICOMP	
BUSINESS BASIC	:BBASIC	:BBASIC
C		:CCXL (no prompt, enter :EOD)
COBOLII	:COBOLII	:COBOLII
COBOLII XL		:COB74XL or COB85XL
CS DOWNLOAD FILES	:RUN CSLIST.PUB.SYS	
CS INTRINSICS	:RUN CSLIST.PUB.SYS	
DS	:RUN DSLIST.PUB.SYS	
DS/X.25	:RUN DSLIST.PUB.SYS	
DTS		:RUN NMMAINT.PUB.SYS
DTC FIRMWARE		:DUI :RUN TERMDSM ST DT n
EDITOR	:EDITOR	:EDITOR
FORTRAN	:FORTRAN	
FORTRAN 77	:FTN	:FTN
FORTRAN 77 XL		:FTNXL
IMF/3000	:IMFMGR (MCHECK)	
IMAGE INTRINSICS	:RUN QUERY.PUB.SYS (VERS)	:RUN QUERY.PUB.SYS (VERS)
KSAM INTRINSICS	:RUN KSAMUTIL.PUB.SYS	:RUN KSAMUTIL.PUB.SYS
MRJE	:MRJE	
MTS	:RUN MPMON.PUB.SYS	
NRJE	:RUN NMMAINT.PUB.SYS	:RUN NMMAINT.PUB.SYS
NS	:RUN NMMAINT.PUB.SYS	:RUN NMMAINT.PUB.SYS
Pascal	:PASCAL	:PASCAL
Pascal XL		:PASXL
QUERY	:RUN QUERY.PUB.SYS	:RUN QUERY.PUB.SYS
RJE	:RJE	
SPL	:SPL	:SPL
SNA LINK	:RUN NMMAINT.PUB.SYS	:RUN NMMAINT.PUB.SYS
SNA/IMF	:RUN NMMAINT.PUB.SYS	:RUN NMMAINT.PUB.SYS
SORT/MERGE	:RUN SORT.PUB.SYS	:RUN SORT.PUB.SYS
TDP	:RUN TDP.PUB.SYS	:RUN TDP.PUB.SYS
VPLUS	:RUN FORMSPEC.PUB.SYS	:RUN FORMSPEC.PUB.SYS

SAMPLE JOB STREAM -- MPE V

```
!job coldinfo,manager.sys;hipri;outclass=lp,1,1
!comment
!comment           This job provides standard information
!comment           for use when placing support calls.
!comment
!comment           Version E.00.00 : MPE V/E
!comment
!comment
!comment
!comment       First, version and IO configuration info
!comment
!continue
!comment
!comment       In place of the SYSDUMP, you could use the
!comment       following:
!comment
!comment       !run sysinfo.prv.telesup
!comment       all
!comment       exit
!comment
!comment       In some cases this listing may be more useful since
!comment       it includes VM address information for each system
!comment       disc.
!comment
!sysdump $null
Y                   <<any changes?>>
                    <<system ID>>
                    <<memory size>>
Y                   <<IO Config Changes?>>
Y                   <<List IO devices?>>
Y                   <<List CS devices?>>
                    <<device defaults>>
                    <<highest DRT>>
                    <<ldev #>>
                    <<max open spoolfiles>>
                    <<list IO devices>>
                    <<list CS devices>>
                    <<terminal type changes>>
                    <<class changes>>
                    <<list IO devices>>
                    <<driver changes>>
                    <<IO configuration changes>>
Y                   <<System Table Changes?>>
                    <<CST>>
                    <<XCST>>
                    <<DST>>
```

```
                <<PCB>>
                <<IOQ>>
                <<DRQ>>
                <<TERMBUFF>>
                <<SYSBUFF>>
                <<SWAPT>>
                <<PRIMSG>>
                <<SECMSG>>
                <<SPEC RT>>
                <<ICS>>
                <<LST>>
                <<UCOP RQ>>
                <<TRL>>
                <<BKPT TABLE>>
                <<max user logging processes>>
                <<max users per logging process>>
Y               <<Misc config?>>
Y               <<List Global RINS>>
                <<delete global rins>>
                <<# of rins>>
                <<# of global rins>>
                <<# of seconds to logon>>
                <<max sessions>>
                <<max jobs>>
                <<default job CPU limit>>
                <<catalog changes>>
                <<softdump changes>>
Y               <<Logging Changes?>>
Y               <<List logging status?>>
                <<status changes>>
                <<logfile rec size>>
                <<log file size>>
Y               <<Disc allocation changes?>>
                <<max directory size>>
Y               <<List volume table?>>
                <<delete volume>>
                <<add volume>>
                <<list volume table>>
Y               <<Virtual memory changes?>>
Y               <<List virtual memory?>>
                <<vol name>>
                <<VM changes>>
                <<max # of spoolfile ksectors>>
                <<# sectors per spoolfile extent>>
                <<scheduling changes>>
Y               &lt;&lt;Segment limit changes?&gt;&gt;
                <<concurrent programs>>
                <<max code seg size>>
                <<max seg per process>>
                <<max stack>>
```

```
                   <<max xds size>>
                   <<max xds per process
                   <<standard stack size>>
                   <<system program changes>>
                   <<system SL changes>>
                   <<dump date>>
!comment
!comment     Now check for Datacomm things
!comment
!continue
!run nmmaint.pub.sys
!continue
!run dslist.pub.sys
!continue
!run cslist.pub.sys
Y
Y
N
!comment
!comment     Write information as to startup
!comment     into COLDDATE.PUB.SYS.
!comment
!file datetime,new;temp;rec=-80,,f,ascii;nocctl
!showjob ;*datetime
!setjcw cierror=0
!continue
!listf colddate;$null
!if cierror=0 then
!editor
text colddate.pub.sys
join datetime (#0/#0)
change 1,"System tape loaded on ",LAST
keep colddate.pub.sys,unn
exit
!else
!build colddate.pub.sys;rec=-80,,f,ascii
!editor
text colddate.pub.sys
join datetime (#0/#0)
change 1,"System tape loaded on ",LAST
keep colddate.pub.sys,unn
exit
!endif
!eoj
```

SAMPLE JOB STREAM -- MPE XL

```
!job coldinfo,manager.sys;hipri;outclass=lp,1,1
!comment
!comment         This job provides standard information
!comment         for use when placing support calls.
!comment
!comment         Version X.00.00 : MPE XL
!comment
!comment
!comment
!comment     First, version and IO configuration info
!comment
!continue
!sysgen
io
lc
ld
lp
lv
ex
lo
sh
ex
mi
sh
ex
sy
sh auto
sh dcc
ex
sh
ex
!comment
!comment     Now check for Datacomm things
!comment
!continue
!run nmmaint.pub.sys
!comment
!comment     Write information as to startup
!comment     into COLDDATE.PUB.SYS.
!comment
!file datetime,new;temp;rec=-80,,f,ascii;nocctl
!showjob ;*datetime
!setjcw cierror=0
!continue
!listf colddate;$null
!if cierror=0 then
!editor
text colddate.pub.sys
join datetime (#0/#0)
```

```
change 1,"System tape loaded on ",LAST
keep colddate.pub.sys,unn
exit
!else
!build colddate.pub.sys;rec=-80,,f,ascii
!editor
text colddate.pub.sys
join datetime (#0/#0)
change 1,"System tape loaded on ",LAST
keep colddate.pub.sys,unn
exit
!endif
!eoj
```

SAMPLE SYSSTART FILE

```
UPDATE
STREAMS 10
STREAM DATAJOB.PUB.SYS
####
COLDSTART
STREAMS 10
STREAM DATAJOB.PUB.SYS
####
```

** END OF FORMATTING **
TDP/3000 (A.04.01) HP36578 Formatter
MON, JUN 6, 1988, 10:06 AM
NO ERRORS
INPUT = EDITOR WORKFILE, TEXT FROM PAPER1
OUTPUT = *HP2680

Optical Publishing: Data Conversion/Preparation for CD-ROM Applications

Jeff Szafransky
Hewlett-Packard
Application Support Division
Mountain View, California

INTRODUCTION

Optical publishing and storage technologies, especially CD-ROM, are changing the way information is being distributed and stored. New commercial CD-ROM applications are being introduced each month, and many internal CD-ROM publishing activities are taking place. Hewlett-Packard is involved with optical publishing and CD-ROM with its HP LaserROM support information subscription service.

Many, when first exposed to this technology, inquire as to the cost of the plastic disc itself. After discovering that the disc is relatively inexpensive, they question the cost of the CD-ROM application. This line of reasoning does not address the underlying effort and expense involved in the production of a CD-ROM - data conversion and preparation.

Taking data and information from its existing format to CD-ROM is a complex multi-step process requiring significant expertise in a number of wide ranging fields. Careful consideration must be given to the option of doing some or all of the data conversion/preparation yourself versus contracting with outside service bureaus. In this paper I will describe the necessary data conversion and preparation steps involved in CD-ROM production.

DECIDING ON A CD-ROM APPLICATION

There are many types of information and many ways of distributing it.

New CD-ROM applications are constantly being introduced, and there are many future applications that have not even been thought of yet. A list of some of the current applications appears below. This list is by no means exhaustive, especially when you consider some of the newer multimedia applications being developed.

Reference Manuals
Catalogs
Training Materials
Product Demos
Software Distribution
Directories
Financial Databases
Business Archives
Census Data
Systems Documentation
Medical Data
Dictionaries and other References
Bibliographic Information

Current distribution methods include paper, on-line services, microfiche, microfilm, magnetic media (floppies and tapes), and the spoken word. An example of the latter would be the traditional training environment with a live instructor.

Selecting CD-ROM to distribute certain types of information involves the analysis of many factors. Needless to say, not all products should be moved to CD-ROM. In making your decision to go with CD-ROM you need to evaluate:

Availability of the Data
Cost for Data Conversion
Storage Capacity Required
Media Costs
Storage Costs
Distribution Costs (CD-ROM vs. On-line, Postage, etc.)
Data Accessibility Requirements
Frequency of Updates
Frequency of Access
Population of target audience
"Market" Demand
Availability of Hardware (PC, CD-ROM Drive, etc.)
Increased usefulness of having data on CD-ROM

SELECTING A DELIVERY SYSTEM

Most CD-ROM workstations are personal computer based with either internal or external CD-ROM drives and a printer. Some systems require a mouse, high resolution graphics displays, and even speakers for audio support. Software includes CD-ROM device drivers and the retrieval software including any necessary decompression or decryption routines. The delivery system requirements are dictated by the application and retrieval software used to access and present the vast amount of data stored on CD-ROM.

Several options are available when deciding on the retrieval software and user interface for the CD-ROM application. You may develop the entire system in-house; develop part of the system in-house using a developers' toolkit from an outside firm; purchase an off-the-shelf system from a vendor; or contract with an outside company for the development of a customized user interface for your application.

Common capabilities of today's CD-ROM retrieval software include windowing, full-text keyword retrieval, browsing, cross references, display and print of both text and graphics, and full help facilities.

See the paper "User Interface Design Methodologies for CD-ROM Information Retrieval" (1) elsewhere in these proceedings for information about the design and development of the HP LaserROM user interface.

DATA PREPARATION OVERVIEW

With an application identified and a user interface selected we may now address the data preparation activities involved with CD-ROM production. See Figure 1 for a high-level graphical representation of the CD-ROM production process. The remainder of this paper will describe each of the data preparation steps in detail.

There are several data preparation companies who are able to perform all of the conversion/preparation steps and provide you with CD-ROM discs. Your expertise, project schedule, staff, resources, and the use of existing tools will need to be evaluated when deciding on the direction to take with the data preparation for your CD-ROM application.

DATA CONVERSION

Data preparation is the most time-intensive and expensive aspect of CD-ROM publishing. In data preparation the one step that has the potential for accounting for the majority of time and effort is data conversion. Data conversion involves the conversion of text and graphics into a format that is compatible with the retrieval and indexing software.

The data conversion efforts required are dependent on a number of things, including:

- CD-ROM production system data input format
- If the data is currently machine-readable or not
- Format(s) of data that is machine-readable
- Existence of conversion tools

Data Input Format

To allow for efficient and effective processing of all types of data to be offered on the CD-ROM, it is necessary to have all of the data enter into the production process in a consistent format.

Given the retrieval and indexing software used in HP LaserROM, we defined a CD-ROM production system data input format. This format consists of ASCII text files that also provide information to the indexing software about the structure of the data so that the correct fields are indexed and the associated control files are built properly. In addition, the text files needed to contain the codes that tell the retrieval/display software how to format the document for display.

We defined a specific graphic file format for the system. We chose the TIFF (Tagged Image File Format) (2) for HP LaserROM graphics. We also specified resolution of the figures and files sizes.

Machine-Readable Format

You can only deliver data on CD-ROM if the data is machine-readable. So, the first step in the conversion process is to make sure all of the data to be placed on the disc is machine-readable.

Text and graphics are handled in similar ways. They both may be scanned or recreated.

Textual data that is in hardcopy only form is scanned in with the use of Optical Character Recognition (OCR) software creating a straight ASCII file. An alternative to the scanning method is to actually re-key the information into a computer system. One reason re-keying may be used over scanning is if the only hardcopy available is of such poor quality that individual characters cannot be recognized by the OCR software.

Graphic images may also be uplifted from paper copies using image scanners such as Hewlett-Packard's Scanjet. Illegible graphics or those that require better resolution may have to be recreated using a suitable graphics package.

Format of Machine-Readable Data

Assuming that we have all of our data in machine readable format we now must look at the differences between the data formats and the input data format required by the CD-ROM production system. It is quite probable that most, if not all, of the data will need to be converted to the CD-ROM production system input format.

There are varying levels of complexity when it comes to data conversion.

If we refer back to our data input format for text we see that we need to know something of the structure of the data. This allows us to quite easily identify those types of data which will be easier or harder to convert. As a general rule, the more structured the data or the more information about the structure the data carries with it, then the easier it will be to convert to the CD-ROM production system input format.

For example, databases are much easier to convert because of their inherent structure. Programs can be written to extract the appropriate data from the database, insert the codes required by the indexing/retrieval software, and create a file that can be used by the CD-ROM production system. The Software Status Bulletin and Product Catalog data in HP LaserROM originated from IMAGE databases, and we used this programmatic extract-and-format approach. On the other hand, reference manuals and other free-format information presents a large challenge to convert to the right format. The problem arises from not being able to identify specific structures within the text that are required by the indexing and retrieval software.

Structure within text refers to constructs such as document name, chapter or section title, publication date, author, or glossary term. Most documentation formatting languages, such as TDP, are concerned with the format of the information rather than the specific structure or components that make up the document. And, the straight ASCII files, which were created as a result of the scanning process described above, contain no structural information at all. In fact, a majority of the formatting (i.e. font) information is lost in this case.

There is, however, a fairly recent development within the publishing industry and Hewlett-Packard documentation development groups to move towards a descriptive markup of electronic documentation. These powerful markup languages, defined by ISO 8879: Standard Generalized Markup Language (SGML) (3), can be used to organize books and articles by identifying necessary structural information. This structural information can be used by various forms of electronic publishing to produce either a hardcopy product or a CD-ROM.

We have developed the HP LaserROM CD-ROM production system to accept SGML-coded documentation as an input format. This documentation is then converted into the standard data input format at the front end of the data preparation process.

The advantages of this approach are many. A single documentation format may be used for both print and CD-ROM. This reduces maintenance and rework, and is analogous to the reuseable software philosophy currently being promoted in the software engineering community.

Since SGML is a fairly new development, the majority of the installed base of reference manuals selected to appear on HP LaserROM is in other formatting languages. The decision was made to convert this documentation into SGML rather than directly into the CD-ROM production system data input format. Since most of this documentation is related to supported products and is currently being maintained, it makes sense to take advantage of the multiple format publishing possibilities of SGML.

The documentation conversion to SGML is a complex issue given the state of the current documentation and the number of formatting languages currently in use within Hewlett-Packard. Some formats are easier to convert than others because there are some formatting languages which better identify the document structure. This allows for at least partial programmatic conversion to SGML. Considerable amounts of hand conversions by skilled converters take place for the worst case formats and for the final tweaking of programmaticlly converted documentation.

These conversions require the expertise of both software engineers and documentation specialists.

Many of the same conversion issues apply to graphics information as well. There are many graphics formats in use today within the documentation development community at Hewlett-Packard. Formats include HP DRAW, Graphics Gallery, EGS/9000, and paste-up artwork of various formats, to name but a few.

Graphic images scanned in using the HP Scanjet produce TIFF files directly, so these images require no conversion. Other scanners produce various formats which require custom programmatic conversion to the necessary TIFF format. Conversion routines for several of the machine-readable formats have also been developed.

As with text, the differing formats and operating environments (e.g. HP3000, HP9000, Vectra) add to the complexity of the conversion effort and accentuate the multi-discipline expertise required.

Fortunately, there is some data that will be placed on the CD-ROM which requires no data conversion. Data referenced but not presented with the text and graphics need only be transferred to an MS-DOS compatible file. Examples of this type of data include software and non-indexed files. This usually represents a small percentage of the overall data to be placed on the disc, however.

As daunting as the data conversion step looks, it is a necessary first step in the data preparation process. Two fairly recent developments in the area of data conversion can be directly attributed to the growth in optical publishing on CD-ROM. The first is the expansion of existing companies into the data conversion business as well as new entities whose sole purpose is to perform data conversions. The second is the increase in the number of data conversion tools being offered by a host of companies.

INDEXING

Given the amount of data a CD-ROM disc is capable of holding (approximately 600 megabytes), serious consideration must be given to how the user will effectively retrieve the information needed. Performing a serial scan through all of the data would take far too long, so a better way to access the data must be employed. HP LaserROM make use of full-text retrieval technology to search through documents.

Full-text retrieval programs developed for CD-ROM applications use inverted indexes to search for data. The indexing software within the CD-ROM production system creates these indexes. The retrieval system index identifies the location of the information on the disc similar to the way a book index lists the location of information in the book.

The indexing method and the retrieval software are integrally connected because the retrieval software must be able to read and act on the index created during data preparation. Both the retrieval software and indexing method are strongly influenced by the type of data and application. For instance, structured databases and full-text applications are indexed in a different way. The primary difference relates to the multiple fields by which the user wishes to access the data in a database. Full-text applications like HP LaserROM require an indexing method that records the location of every word in every document. This allows the retrieval/display software to highlight the search word when displaying the document, as well as provides the information necessary to perform phrase and proximity searching.

Other components of the document such as chapter titles, authors, publication dates, and the like may also be indexed. Cross-reference indexes are also created in this step.

The indexing step is complicated by the fact that there are parts of the data that do not require indexing. Display enhancement codes are an example of this type of data. The indexing must be capable of being turned on and off automatically during the processing of the data.

Another example where the indexing is temporarily inhibited is for words that you do not want to index, also known as stopwords. Typically, they are (1) articles and prepositions (a, the, of, for, etc.) or other words that add no value to the search or (2) words that appear in so many of the documents that a search on one of these words does nothing to help narrow down the number of qualifying documents. These stopwords are entered into the indexing process via an excluded word list/stoplist file.

Full-text retrieval and its associated indexes require a great deal of disc space. The overhead for certain types of data can be as much as 100 percent. The indexing step, though automated, can also take considerable time depending on the processor used. Despite this, full-text retrieval is the best way to access large amounts of unstructured text and structured database data. So, the indexing must go on!

COMPRESSION/ENCRYPTION

Data compression and encryption of the data on CD-ROM is another very large area which I will only touch on briefly here. I will not go into all the possible methods of data compression or encryption, just some of the reasons for performing these activities and issues involved.

The 600 megabyte storage capacity of today's CD-ROM seems almost infinite to most users, and the technology is progressing so as to make even this figure seem small. But, if you consider that a single graphic image may require more than a megabyte, and that the overhead of indexes could be as much as 100 percent, then you begin to think of how to save some space so you can get your entire application on one disc! This is where data compression enters the picture.

Compression is a process that converts data into a form that requires less space. There are numerous compression schemes for both text and graphics, and these compression schemes may be performed in either hardware or software. Compressed data by definition requires decompression at some time, and this can also be handled in hardware or software.

Information distribution using CD-ROM eliminates the data vulnerability due to unauthorized access over networks and telecommunications systems, but some protection may still be required to protect the data on the CD-ROM disc from unauthorized use. CD-ROM applications can involve highly sensitive data such as defense information or internal company documentation. Or a company may just want to make it harder for someone to do a mass-downloading of all of the information on the CD-ROM. Data encryption is one method of securing data on CD-ROM. Many of the issues involving implementation of data encryption for CD-ROM are the same as those for data compression.

Issues to be considered when evaluating compression or encryption for your CD-ROM application include:

- Is it necessary? (Do you require the extra space or the data security?)
- Is the data a good choice for compression? (How much space will you save?)
- Does the indexing software support the compressed/encrypted data?
- What is the impact on the CD-ROM production system?
- Does the retrieval software support the compressed/encrypted data?
- Is special software or hardware required?
- What is the effect on the retrieval system performance?

You must also evaluate the various compression and encryption methods and select the method that performs best with your application.

DATA LAYOUT

At this point in the CD-ROM production process all of the files are ready and in the form required by the delivery system: data is formatted, indexes are built, graphics are scanned, the data may be compressed and/or encrypted, and all control and support files are present. Data layout is the process of arranging your files and directories in the exact order and location as they will appear on the CD-ROM. The result is known as the disc image, and the final CD-ROM will precisely reflect its structure and content.

The layout of the data onto the disc image usually only needs to be done once, and then each subsequent disc production follows the same layout script. The layout of the data depends on the application, and the primary factors are access time and overall performance of the delivery system. Due to the relatively slow access times of the CD-ROM drive it is necessary to pay particularly close attention to the placement of files on the disc. Generally speaking, files that are accessed together often should be placed in close proximity to each other. Knowing in detail the specific operations of the delivery system allows for a knowledgeable data layout.

Data layout involves the following operations:

1. Collect all files to be placed on the CD-ROM
2. Determine disc directory structure and which files reside in which directory
3. Create the necessary directories
4. Transfer files into their appropriate directories in the disc image

The underlying logical format of most CD-ROM discs is the High Sierra Format or the ISO 9660 format, which is the international standard developed from the High Sierra Format. Most commercially available data prep systems and virtually all service bureaus support these formats.

EMULATION

Once the disc image is made and while the data is still residing on the winchester disc, it is necessary to test the data integrity and simulate how the data will operate with the delivery system. Checking the data integrity involves activities such as making sure all files exist, are the right size, and that they can be opened/executed.

Systems exist that allow for fairly rigorous emulation (or simulation) of the CD-ROM application using the disc image. Some of these systems go so far as to exactly duplicate the seek and read times of many of the CD-ROM drives currently available. This system emulation is the only satisfactory way of verifying that everything has come together correctly and that the application will work once it is placed on the CD-ROM. Waiting until the CD-ROM is created is a risky and potentially expensive alternative.

Checking the results of the emulation may reveal problems with the disc image. Files may be missing or in the wrong location; files may be in an incorrect format; a problem with one of the previous data preparation steps may be uncovered; or the layout of the data may need to be modified.

If a change is required, then we need to go back to the appropriate data preparation step and fix the problem, work through the subsequent steps, and then the emulation is performed again. If the emulation is successful, then we proceed to the next step in the process: premaster tape generation.

The final step in the formal CD-ROM production system is the generation of a premaster tape. The disc image is copied to an ANSI labelled 9-track tape according to one of the CD-ROM mastering plant's input specifications.

PREMASTERING/MASTERING

This section, while technically not part of the data preparation process, is included for completeness. I want to briefly describe the steps involved in the actual CD-ROM disc production.

Very few companies will actually produce their own CD-ROM's. The technology and economics make it much more feasible to go with one of the existing mastering plants.

When the 9-track tape arrives at the mastering facility it is scanned for readability and conformance to format specifications. Mastering facilities do not check logical format, disc layout, or disc directories. This is the responsibility of the group producing the premaster tapes. Whatever data is on the tapes will be mastered to the CD-ROM.

Premastering is the process where the data is transferred from tape to hard disc and the error detection and error correction codes are added to the data. Header information and sync bytes are also added to the data blocks.

Mastering is the process where a master disc (usually made of glass) is produced. The data bits are etched onto the glass using a laser and are represented by pits and lands. A negative image of this disc (usually made of metal) is produced and this becomes the "stamper" which is used to create the CD-ROM discs.

Compact discs are made of clear polycarbonate plastic. Replication of the discs is performed by an injection molding process using the metal stamper to emboss the data patterns into the plastic. Reflective silvering is added covered by a protective plastic coating, and then labels are printed on the discs. Various amounts of packaging may be added at the end of the process. This usually includes the loading of the CD-ROM into either a jewel box or CD-ROM caddy.

The quality control processes within the CD-ROM mastering facilities is quite impressive. Some mastering facilities are so confident about their quality control processes that they guarantee the CD-ROM discs you get will exactly match the data you sent them on tape. Statistics show that a defective CD-ROM is usually the result of incorrect data on the premaster tapes. This highlights the necessity for ensuring that the data sent to the mastering plant is 100 percent accurate.

CONCLUSION

The intent of this paper was to give an overview of what is involved in the creation of a CD-ROM application with regards to data conversion and data preparation. All of the data preparation steps described in this article may be performed either internally or through an ever growing number of service bureaus.

Most CD-ROM software developers are publishing information directly or providing contract development services. In either case the organization most knowledgeable about the customer and the data (the information provider) gives up control of product development and sometimes marketing. As CD-ROM technology has gained acceptance, organizations are seeking to gain control of the development process internally. Providing information managers with their own internal CD-ROM publishing capability enables them to tailor the technology to their business and customers for maximum competitive advantage. Through careful analysis of your application, expertise, schedules, and resources you will be in the best position to determine how to implement an optical publishing solution within your company.

But, let us get back to the original question: "How much does that disc cost?". I would have to answer with something like the following.

"The cost per disc is determined by the following costs:

Data Collection
Data Conversion
Data Preparation
Premastering
Mastering
Disk Duplication
Packaging

plus

Retrieval Software (sometimes charged per disc or CPU)
Documentation
Technical Support
Marketing
Copyright Fees
Licensing Fees

not to mention the cost of development for the retrieval software and the data preparation system, hardware for the data preparation system, ... "

Bibliography

1. Ferguson, Greg. "User Interface Design Methodologies for CD-ROM Information Retrieval" Hewlett-Packard. Presented at 1988 International HP Users Group Conference, May, 1988.

2. Andrews, Nancy and Fry, Stan. "TIFF: An Emerging Standard for Exchanging Digitized Graphics Images". Microsoft Systems Journal. July, 1987.

3. International Organization for Standardization (ISO). ISO 8879-1986(E). Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML). International Organization for Standardization, Switzerland, 1986.

Other Useful References

Lambert, Steve and Ropiequet, Suzanne (ed.). CD-ROM: The New Papyrus. Microsoft Press, Redmond, Wash., 1987

Ropiequet, Suzanne (ed.), Einberger, John and Zoellieck, Bill. CD-ROM: Optical Publishing. Microsoft Press, Redmond, Wash., 1987

Roth, Judith Paris (ed.). Essential Guide to CD-ROM. Meckler Publishing, Westport, Conn., 1986.

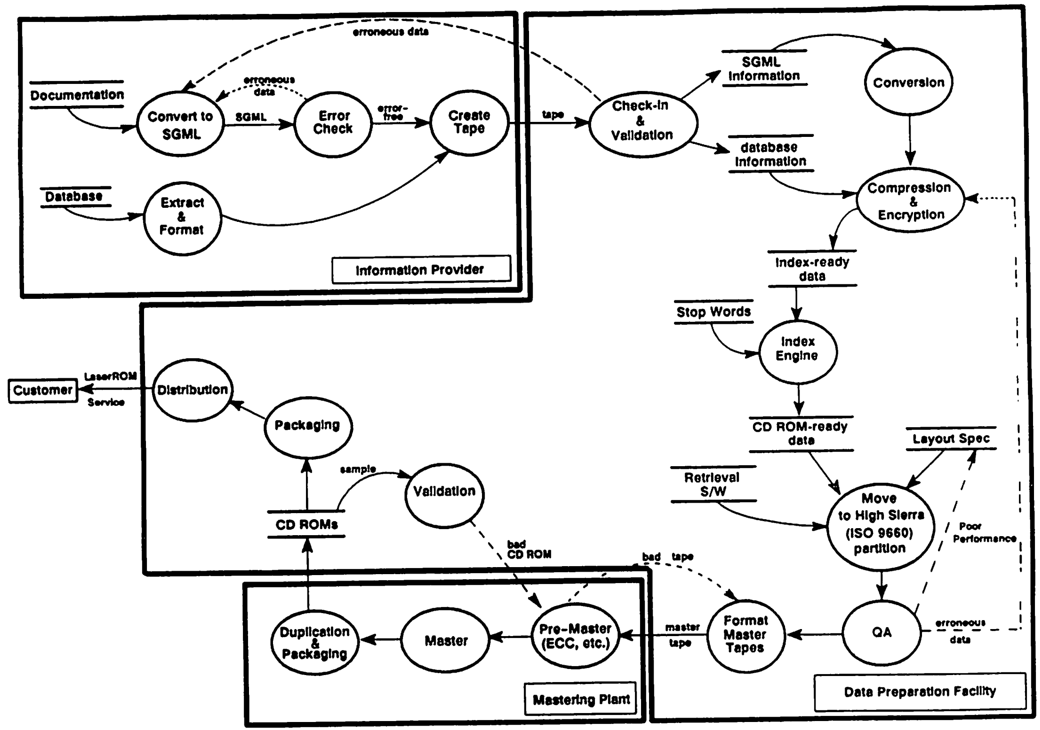


Figure 1 — CD ROM Production Process

BUILDING A COMPLETE DISASTER RECOVERY PROGRAM

Gregory A. Morris and Norman Moyer

Hewlett-Packard Company
100 Mayfield Avenue
Mountain View, California 94043

ABSTRACT

As businesses come to rely more and more on their computing capabilities, the cost of downtime in the event of a computer disaster is becoming increasingly severe. Businesses face mounting pressure from external auditors and internal staff to prepare for the possibility of a computer disaster. With the help of outside consultants, companies are developing detailed disaster recovery contingency plans and then supporting these plans with backup computer hardware. Unfortunately, companies often cannot go to a single vendor that meets both their planning and hardware backup needs. And it is rare indeed for a vendor to take the additional step of offering a program to expedite reccvery of a disaster site.

Focusing on the dilemma companies face in putting together a complete disaster recovery program, this paper reviews pivotal disaster recovery issues and considerations. HP Disaster Recovery Services are presented as one solution to the problem of implementing a complete disaster recovery program.

SITUATION ANALYSIS: THE NEED FOR DISASTER RECOVERY SERVICE

The importance of computers in business has grown dramatically since the days when organizations looked to computers simply to facilitate numerical computation. Today, computers are fundamental to every aspect of business, from financial management to order processing to manufacturing. In light of this increased reliance on computers, there naturally is increased concern that computer operations be able to continue under any circumstances. A complete disaster recovery program can significantly reduce the impact of a disaster on a company's normal operations.

There are several specific reasons that companies are investing in disaster recovery programs:

- **Financial Impact.** Computer disasters are costly. Every day that the computers do not function can mean orders lost, orders not filled, or, at the very least, implementation of cumbersome manual procedures.

- **Business Survival.** Losses can mount so rapidly that a company's only viable option is to discontinue business.

- **Audit Requirements.** Auditors are demanding that computer-dependent entities have testable disaster recovery plans.

- **Government Regulation.** Government organizations are being forced to develop disaster recovery programs. Recent legislation sponsored by Senator Alan Simpson (R-Wyoming) specifies initial guidelines for computer security for all federal agencies and subcontractors to the federal government. The legislation, which is expected to be the first of several bills involving data processing security, includes a discussion of disaster recovery planning and hardware backup. Increasing legislative pressure is motivating private industry as well.

- **Management Liability.** The Foreign Corrupt Practices Act (1977) established that management can be held criminally liable for failure to protect resources critical to a company's operations. Thus, management of computer-dependent companies are legally responsible for establishing disaster recovery provisions.

Any computer-dependent company, in any industry, needs to understand the potential impact of a computer disaster and act to minimize the impact. However, the need for a disaster recovery program is most critical in a few specific industries.

The **financial services** industry -- banks, insurance companies, investment houses, etc. -- has an especially great need for disaster recovery protection. The industry is highly automated and encounters the strictest of audit requirements. **Manufacturing** companies need protection because computer-aided manufacturing can literally grind to a halt because of a disaster, resulting in the shut down of expensive manufacturing processes. Companies focused on **distribution** have completely automated shipment coordination, and risk losing valuable business if a computer disaster paralyzes this function. The **health services** and **petro-chemical** industries also are beginning to invest heavily in disaster recovery programs. In addition, many other companies rely on computers for billing, payroll, and other critical tasks.

A COMPLETE DISASTER RECOVERY PROGRAM

Recovery from a disaster is not complete until all normal computer operations have been restored at the disaster site or at a replacement computer facility. To enable this to happen, a complete disaster recovery program explores and defines the actions to be taken before, during, and after a disaster.

Disaster Recovery Planning. A disaster recovery program begins with a detailed contingency plan for interim processing and actual recovery of a disaster site. Decisive action based on advance planning is the first step toward minimizing the impact of a disaster.

The objective of a recovery plan is to enable a company to resume processing of critical computer applications as quickly as posible, and not necessarily to duplicate the normal operating environment. The plan documents emergency procedures for resumption of critical data processing functions and provides for longer-term recovery of non-critical computer operations.

There are three approaches to developing a disaster recovery plan. First, a company can write its own plan, using its own common sense and publicly available literature and information. Although this is a straightforward approach to planning, companies that do their own planning are not able to benefit from the expertise that specialists can provide. A professionally prepared recovery plan almost always is more thorough than a plan written by untrained personnel.

Second, a company can obtain assistance from a planning vendor, in order to write a thorough plan on its own. Some vendors offer training in disaster recovery planning and provide comprehensive planning methodologies designed specifically for a given computing environment. This planning alternative requires substantial time commitments from company personnel, but helps ensure that a company's planning process identifies and addresses all of its particular needs. Furthermore, because a company still retains responsibility for writing its plan, the company can readily modify or update the plan without additional assistance. Ongoing testing and maintenance of a recovery plan is the only way to ensure having a workable plan when it is needed.

Third, a company can hire professional planning consultants to write a plan. This alternative can be quite costly and forces a company to rely on a consultant to update the plan. However, having a consultant write a recovery plan may save time and make company employees available for other tasks.

Hardware Backup. Hardware backup services provide backup computer hardware on which a company can continue its computer operations during a disaster. The process of documenting a disaster recovery plan typically identifies precisely what operations need to be backed up and what hardware is required to accomplish this.

There are several categories of hardware backup service. The most common service provides a stand-by hotsite -- a computer room with a computer system, peripherals, and telecommunications capabilites. Although some portion of a company's MIS staff may have to travel to reach the hotsite facility, having access to a full-scale computer facility gives a company one of its best hopes for duplicating normal operations.

Mobile backup is a popular alternative to the hotsite. Under a mobile backup scenario, a vendor sends a small computer room located in a trailer to a predetermined location. Since mobile backup must be offered on a small scale, this backup alternative generally works best for smaller operations and in countries where having to transport a mobile facility over long distances is not a major concern.

Companies with access to an alternative site that can serve as a computer facility during a disaster may prefer express delivery of loaned hardware. Under this option, the company contracts with a vendor for loaned hardware until replacement hardware is available. Express delivery is appropriate for customers who need a relatively small amount of equipment and minimal technical assistance.

Many companies write disaster recovery plans without assistance; many others seek to provide their own backup program. Some companies are so computer dependent that they have their own backup computer facility. Since the full cost of the facility is borne by one company, this alternative is practical only when a company has especially unique equipment requirements or when data security is a major issue.

The reciprocal agreement, whereby two companies agree to share their computer equipment in the event that one experiences a disaster, is a backup alternative that does not require the services of a commercial vendor. Reciprocals are advantageous because companies can minimize the cost of backup and still have access to a large-scale, fully operational computer facility. One drawback is that companies with reciprocal agreements historically have not reserved sufficient capacity for their reciprocal partners. A related drawback is that testing of recovery procedures usually involves disruption of normal operations for two businesses.

Disaster Site Restoration. Once a disaster strikes, a company uses its disaster recovery plan not just to implement interim procedures, but also to restore the disaster site. Vendors are available to salvage equipment and clean disaster sites; however, a company's primary focus typically is on obtaining replacement hardware from the equipment manufacturer. The manufacturer can alleviate this conflict by taking a strong leadership role in the restoration effort.

THE SERVICE DILEMMA

To be complete, a disaster recovery program must consider recovery planning, hardware backup, and disaster site restoration. Preparing for these three aspects of disaster recovery is essential for full protection before, during, and after a computer disaster.

Today, companies are able to turn to a variety of specialists to build their recovery programs. These specialists typically focus on one aspect of disaster recovery -- planning, backup, or salvaging. They develop expertise in their area of focus that gives customers confidence in the services being provided. However, although computer users must invest in all areas of disaster recovery, there is no natural link between the services provided by multiple specialists.

A single vendor that offers planning, hardware backup, and disaster site restoration assistance enables a company to have an integrated recovery program and develop a well-defined partnership with a single recovery services vendor. This may not be of value to the company whose operations are relatively uncomplicated, easily understood, or easily supported during a disaster. For many organizations, though, no price can be placed on the added security that is guaranteed by the consistency of working with just one vendor.

The multi-vendor dilemma is understandable because the disaster recovery industry is young. Many vendors are small, regional players. Few standards have been established. Nevertheless, heightened management awareness of the industry and an ever-greater need for recovery services are certain to bring about standardization of services. The Hewlett-Packard Company is an example of a company moving in this direction by this year becoming the most-recent major computer manufacturer to offer a complete array of recovery services.

HP DISASTER RECOVERY SERVICES

HP Disaster Recovery Services presently consist of two products. Although the products are available separately, together they provide a complete disaster recovery program.

HP Disaster Recovery Planning. HP Disaster Recovery Planning supplies the knowledge and tools necessary to develop, document, and maintain a disaster recovery plan for HP 3000 users. Based on a planning methodology customized for the HP 3000 environment, the service provides a structure for a company's recovery plan. A PC-based recovery plan template enables customers to write a plan that conforms to site-specific policies, practices, and conditions. To enhance the value of the planning tools HP provides, HP Disaster Recovery Planning includes two days of on-site instruction on HP's planning methodology. Recognizing the value of involving the customer in planning, HP also includes a project management guide to facilitate plan development.

The methodology covers identification of critical applications and alternate processing procedures, administrative procedures during a disaster, management of computer inventory during a disaster. Procedures for testing and maintaining a recovery plan assist auditors in determining the soundness of customers' recovery programs.

HP Backup. HP Backup provides access to an HP 3000 computer facility in the event of a system disaster. The customer travels to a facility dedicated to disaster recovery to run essential computer operations. The facility is equipped with HP 3000 systems and peripherals, an additional empty computer room for customer use, and telecommunications capabilities that HP tailors to meet individual needs. Customers receive six days a year at the recovery facility to rehearse and refine their recovery procedures.

As the equipment manufacturer, HP is in a unique position to offer technical support to HP Backup customers and provide disaster site restoration assistance. HP Backup personnel are on site and on call throughout the recovery process -- both at rehearsals and during actual disasters -- to help translate customers' recovery plans into action, answer operational questions, and facilitate access to special additional equipment. At the disaster site, HP works with the customer in all activities from performing a preliminary inspection of the disaster site to expediting shipment of replacement hardware.

CONCLUSION

The disaster recovery industry is still in its infancy. As the industry matures, vendors are likely to consolidate their services, as the necessity of a total disaster recovery solution becomes clearer. The total recovery solution takes a company from planning of recovery procedures to backing up computer operations during a disaster to re-establishing normal operations. The total solution means total protection of a business. Given the growing importance of computers in day-to-day business operations, HP Disaster Recovery Services is one of the first of what probably will become a variety of disaster recovery programs from computer manufacturers.

Gregory A. Morris is presently Systems Support Product Manager for the Product Support Division at Hewlett-Packard in Mountain View, California. In this capacity he is responsible for the development and marketing of disaster recovery backup hardware products. Gregory joined HP in 1987 and holds a Bachelor of Arts degree in Economics at Princeton University and a Master of Business Administration degree from Stanford University.

Norman Moyer is presently a Product Manager for the Application Support Division at Hewlett-Packard in Mountain View, California. In this capacity he is responsible for the development and marketing of disaster recovery planning products. Norm joined HP in 1985 and holds a Bachelor of Science degree in Economics from the Wharton School, University of Pennsylvania.

TITLE: Finally: Performance Technology!

AUTHOR: Tony Engberg

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 2026

HP ADVANCENET: AN OVERVIEW

KARYN MASHIMA

HEWLETT-PACKARD COMPANY

CUPERTINO, CALIFORNIA

INTRODUCTION

Most HP networking users are focused on those products and services that meet their particular needs. The purpose of this paper is to give a wider overview of HP AdvanceNet, the company's networking strategy, that describes how Hewlett-Packard is attempting to meet the tremendous diversity of business needs that our customers have.

It's important to remember that HP AdvanceNet is a strategy, not an architecture. The goal of HP AdvanceNet is to give users a competitive advantage through comprehensive and flexible multivendor networking.

Why the emphasis on multivendor networking? Because our experience and customers tell us that the ability to connect the computers and systems of different vendors is critical to competitiveness. There are many reasons why multivendor environments have become so widespread -- the inability of any single vendor to meet the diverse needs of different businesses and industries, growth through merger and acquisition, and different decision makers -- and the clear trend is for the growth, in prevalence and complexity, of such environments.

The HP AdvanceNet strategy has three key components:

--strong support of the two dominant architectures in networking: OSI (Open Systems Interconnection) and SNA (Systems Network Architecture), IBM's proprietary networking architecture.

--focused networking solutions for distributed information processing in business offices, engineering, and manufacturing environments;

--quality products, consulting, and customer support.

HEWLETT-PACKARD'S NETWORKING EXPERIENCE

Hewlett-Packard didn't develop its strategy in a vacuum. HP AdvanceNet is based on extensive experience with both our own corporate network and the networking needs of hundreds of organizations.

HP's first networking products were introduced in 1973. Today there are more than 300 HP AdvanceNet products, with more than 60,000 networked nodes installed worldwide. There are 81 HP networking customers in the top 100 companies of the Fortune 500.

Customers benefit from HP's experience with its own worldwide, multivendor network. Hewlett-Packard currently employs 82,000 people in 70 nations. The 438 HP sites use 2,5000 host computers and 70,000 workstations, among which are included PCs, terminals, and engineering workstations.

Some 130 billion characters are transferred every month along this network. We use both public and private X.25, with 17 backbone switches, 400 standalone switches and 2 management centers, one at corporate headquarters in Palo Alto, California and the other in Geneva, Switzerland.

That's an extraordinarily complex network, and Hewlett-Packard depends on it for the daily conduct of business.

STANDARDS: THE FOUNDATION OF HP ADVANCENET

Standards are central to HP AdvanceNet because they are the best way to protect users' current investments while allowing for future growth and change.

HP was the first major computer vendor to replace its proprietary networking services and protocols with their OSI equivalents. Where OSI standards have not reached full maturity, HP AdvanceNet supports de facto standards such as TCP/IP and ARPA.

Hewlett-Packard employees currently work on more than 40 task groups and committees involved in the establishment of OSI standards. In addition, the company was a co-founder of COS (Corporation for Open systems) a non-profit, multinational consortium of users and vendors that is developing conformance tests for OSI products. Hewlett-Packard recently became one of the first United States companies invited to join SPAG, the Standards Promotion and Application Group, the European counterpart of COS.

FIVE FOCUSED SOLUTIONS

HP AdvanceNet's five focused solutions are designed to meet networking needs in the primary business areas of both service and manufacturing companies. They are based on the internal automation environment and the need to communicate with external sites.

Each solution is comprised of modules, which address different networking needs commonly found in that environment.

The Business Office solution delivers both local and long distance transaction processing, PC integration that enables users to share resources such as printers and plotters, and voice/data/text integration in both local and wide area networks.

HP StarLAN and HP StarLAN 10 are the heart of the Business Office solution. They are based on the IEEE 802.3 industry standard for local area networking, and they enable users to connect PCs and minicomputers over standard telephone wiring (unshielded twisted-pair). HP StarLAN runs at 1 Mbps, while HP StarLAN 10 operates at 10 Mbps.

HP AdvanceNet for Regional Sales and Service provides four communications alternatives for connecting branch offices to each other and to the regional office. The alternatives are:
--dial-up phone lines: the smallest up front investment and the most economical solution for offices requiring low-volume batch communications.

--public X.25: best suited for batch and interactive applications with low to medium data volume.

--leased lines: good for batch and interactive traffic with higher data volume.

--private X.25: provides maximum security and flexibility for regional communications.

The HP Engineering solution addresses a key issue in this environment: the need to develop higher quality products using a shorter design cycle and fewer engineers.

The proliferation of workstations in engineering necessitates strong connectivity to other PCs, minis, and mainframes. In addition, the popularity of UNIX as an operating system is an important trend.

The HP solution delivers multivendor connectivity using the IEEE 802.3 LAN as the backbone network and the high-performance link for clustered workstations. ARPA and Berkeley services, Network File System (NFS), and the use of UNIX and MS-DOS de facto standards are also major elements of this solution.

An effective network for manufacturing has to integrate all functions fully, including planning and control, financial systems, production processes, manufacturing engineering, and product design.

HP's CIM (Computer Integrated Manufacturing) solution uses de facto and OSI standards, particularly MAP (Manufacturing Automation Protocol). A key to the HP CIM solution is the use of subnetworks that enables users to integrate their processes in a planned, non-disruptive manner.

The HP Private Packet Network (PPN) is the centerpiece of the Company-wide solution, and it's augmented by a range of point-to-point and dial-up alternatives.

Based on the X.25 international standard, the HP PPN delivers strong multivendor connectivity, flexibility to tailor and change a network high reliability and security, and extensive network management and control.

It's common for organizations to combine private and public X.25 connections for the most cost-effective solution.

NETWORK MANAGEMENT

The goal of HP's network management strategy is to enable users to create and manage private networks through all phases of the network life cycle.

HP's hardware and software offerings are based on the six functional areas of network management as defined by ISO: Common Management Information Services/Protocol (CMIS/P) as well as the Specific Management Information Services (SMIS), which are fault, performance, accounting, configuration, and security management.

Hewlett-Packard recently announced HP Openview, a comprehensive network management solution for both local and wide area networking that is based on the OSI architecture.

INDUSTRY-LEADING SUPPORT

Customer service and support are important differentiators for HP AdvanceNet. The annual Datapro customer survey has ranked HP support #1 in the industry for the last five years running. We're continuing to add to our support offerings, facilities and staff as the needs of customers evolve and change.

Our three wide area networking centers are good examples of this. Located in Atlanta, Bristol (England), and Singapore, these centers provide complete range of network management and control services, including total 24-hour-a-day management of private X.25 networks.

CONCLUSION

Scalability and flexibility are important benefits for the decision maker who chooses HP AdvanceNet. This strategy, based on strong support for industry and de facto standards, provides multivendor networking that grows as an organization grows while protecting current investments in equipment and training. In addition, HP AdvanceNet delivers industry-leading service and support.

HP AdvanceNet For Business Office Solution

Felicia Choy
Hewlett Packard
19091 Pruneridge Avenue
Cupertino, CA 95014

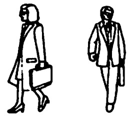
Business office environment is typified by the diversified end users needs ranging from managers to office workers and expertise with frequent personnel moves and changes to accommodate growth, to enhance communciation, cut costs, and increase productivity. As computing move from centralized to more and more distributed and increased use of personal productivity tools has made PC proliferation an epedemic. Distributed computing will also require more sophisticated application development environment, The increasing numbers of companies in a network makes network maintenance a more complex task.

The challenge to meet the networking needs of this dynamic environment is to provide a very flexible, transparent and convenient access network that allows easy sharing of information. This would mean close integration of PC's, terminals, and minis; resource sharing throughout the network; solvable networking solutions from the systems perpective, distributed application and data environment is needed to meet the demand of this environment. Multi-vendor communication and high availability of the network and system is also a necessity.

Business Office

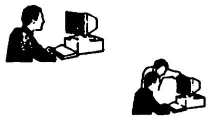
Networking Environment

Users:



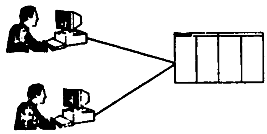
- **Wide range of needs & expertise**
- **Frequent adds, moves & changes**

Workgroup:



- **Proliferation of PC's & personal productivity tools**
- **Use of terminals**
- **On-going growth**

Systems:

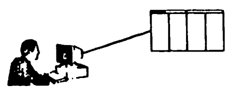


- **Increasing network maintenance complexity**
- **Sophisticated application development and version control**

Business Office

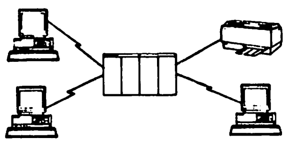
Networking Needs

Users:



- **Ease of use**
- **Transparent, flexible access to information (systems)**

Workgroup:



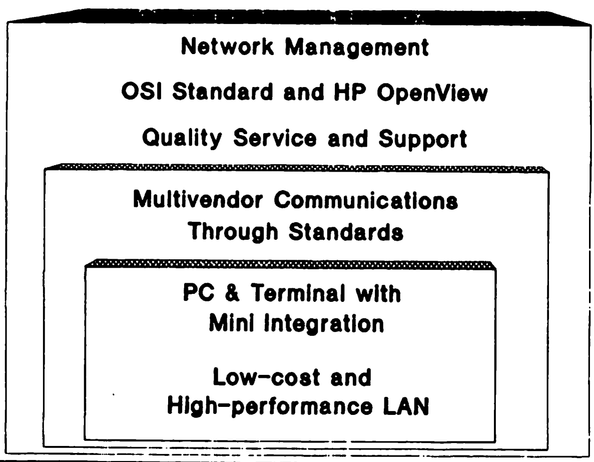
- **PC and terminal/mini integration**
- **Network-wide resource sharing**
- **Scalable networking solutions**

Systems:

- **Multivendor communications**
- **High availability of network/system**
- **Distributed application services**

The Hewlett-Packard AdvanceNet Business Office Network Solution is the solution to the business office networking needs. The low cost and high performance LAN with full PC/mini/terminal integration addresses the needs of frequent moves and changes and growth. It also offers networking services based in industry standards (OSI) and other de facto standards. This facilitate multi-vendor communication. It also offers centralized network management based on the emerging OSI standards and HP's openview.

Business Office Network Strategy



HP ADVANCENET
BOS06

HEWLETT PACKARD

An individual end-user can access the HP 3000 application through a range of workstations connected either locally or remotely. The terminals or PC's are connected to the HP 3000 through the distributed terminal controller (DTC) for the new 900 series of HP 3000 companies and the Advanced Terminal Processor (ATP) for all other HP 3000. The HP AdvanceLink software provides terminal emulation and file transfer capability for the PC's. Remote PC's can also access the HP 3000 application software via the HP serial Network sftware.

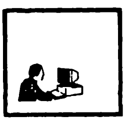
All these provides the individual end user with a high degree of integration for the PC.

HP AdvanceNet for the Business Office

End User Solution - Individual

- A range of workstations
  - Terminal
  - Low-cost PC
  - 386 based
- Personal software
  - Word processing
  - Decision support
  - Graphics
  - Personal data base

Individual



- Personal computing

HP ADVANCENET
BOS08

HEWLETT PACKARD

For groups of PC users, HP StarLan provides convenient connection. This is a twisted-pair wiring scheme with either 1 mbps or 10 mbps based on IEEE 802.3 standard. A workgroup PC LAN provides disc and file sharing, printer sharing, and spooling, as well as, plotter sharing and spooling on a PC server.

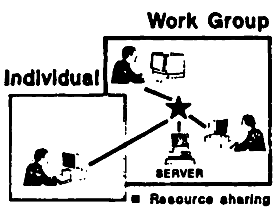
HP AdvanceNet for the Business Office

End User Solution – Workgroup

- **HP STARLAN – the office network:**
  - **Uses the regular phone wiring**
  - **Eases installation, adds, moves and changes**
  - **1 and 10 megabits**
  - **Standard-based IEEE 802.3**
  - **Basis for data/voice evolution**

- **Applications**
  - **Personal productivity**
  - **Workgroup intercommunication**
  - **Local disc sharing**
  - **Local printer sharing**

- **A range of PC servers from 286 to 386 based PCs**



HP ADVANCENET
BOS09

HEWLETT PACKARD

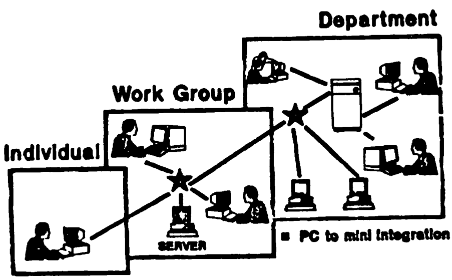
For a large department of end-users, this alternative provides the ability to connect an HP 3000 located either in the department or in the data center with terminals, printers, and personal computers. It enables this entire department to connect the site backbone to access additional information and resources located in the data center or other work groups and departments throughout the entire network.

The center of the department connectivity is the HP 3000. Work groups of PC's are connected to the HP 3000 using StarLan. PC users can access the resources of HP 3000 transparently, as if the HP 3000 printers and discs were local PC peripherals, by running Business System Plus or Resource Sharing software on the HP 3000 and StarLan user link software on the PC.

HP AdvanceNet for the Business Office

End User Solution - Department

- A scalable, cost effective PC to mini integration
- HP STARLAN - the office network:
  - Support HP 3000 mini computer
  - Support HP Personal Productivity Center
  - Distributed applications between PC's and minis
- Applications
  - File back-up and restore
  - Electronic mail services
  - Information access
  - Print Central
  - Resource sharing
  - Distributed applications



HP ADVANCENET
BUS10

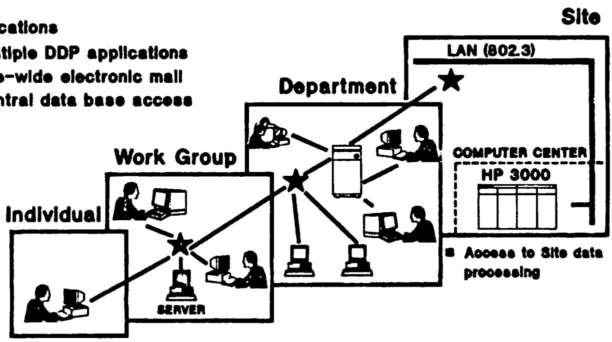
HEWLETT PACKARD

The computer center solution consists of products that enable several HP 3000's residing in the computer center to be linked together and to pass data between applications or between users and applications. It offers Network Services for end users and programmers. It also provides capabiliities for the system or network administration to manage and control the computer center resources. The computer center network is based on the industry standard IEEE 802.3 LAN, and the transport protocols are based on the industry standard TCP/IP Arparnet protocols. HP's committment to industry networking standards provides the basis for multi-vendor communications within the computer center.

HP AdvanceNet for the Business Office

Computer Center Solution

- A scalable, cost effective PC to mini integration
- HP STARLAN – the office network:
  - Connect into the site LAN backbone
- Applications
  - Multiple DDP applications
  - Site-wide electronic mail
  - Central data base access



HP ADVANCENET
BOS11

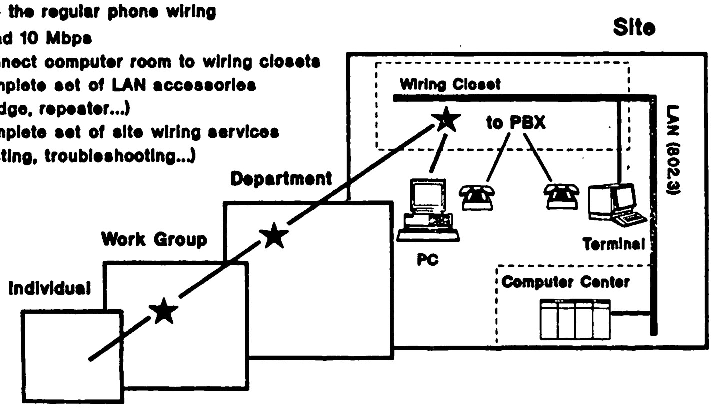
HEWLETT PACKARD

Unshielded twisted-pair cable is the logical choice for business office environments as it is requires for the telecommunication system and is often already in place. It offers a flexible ecomomical alternative to coaxial cable. HP offers a wide range of links that are supported over unshielded twisted pair, e.g. low speed asynchrous links (ATP); 1 mbps StarLan and 10 mbps StarLan connections.

HP AdvanceNet for the Business Office

SiteWire Solution

- **HP STARLAN - the office network:**
  - Use the regular phone wiring
  - 1 and 10 Mbps
  - Connect computer room to wiring closets
  - Complete set of LAN accessories (bridge, repeater...)
  - Complete set of site wiring services (testing, troubleshooting...)



HP ADVANCENET
BOS12

HEWLETT PACKARD

In the business office environment, there is often a need to share information with systems located in other remote data centers within the company such as corporate headquarters. These data exchange capabilities are needed for end users who access remote applications and data bases and for software processes running over multiple systems. There may also be a need or HP 3000 users to communicate with an IBM mainframe at corporate headquarters to updata a data base or exchange electronic mail with corporate IBM users.

The company--wide Access Solution describes the way HP systems can interconnect with remote HP or IBM systems across the backbone network. The alternatives depend upon which type of backbone network is installed as the company-wide network.

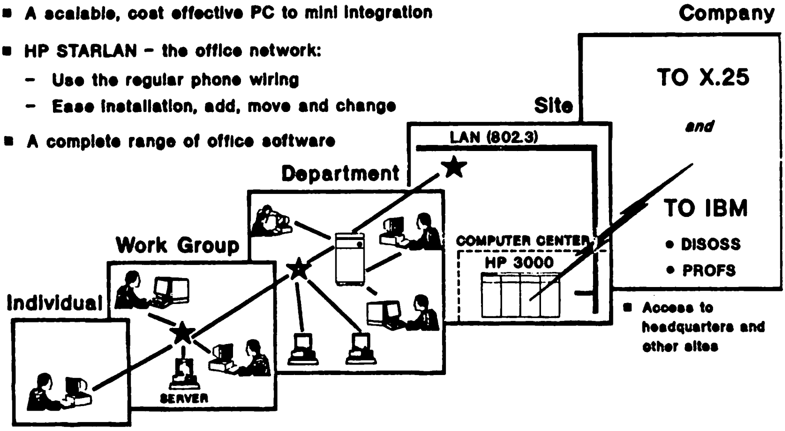
For access to either a public or private X.25 network, HP offers two types of communications. HP 3000 computers can communicate with remote HP data centers using either 1) direct X.25 system access or 2) X.25 system access through a LAN gateway. If it is necessary for HP computers to communicate with IBM systems, then the alternatives are 1) an HP 3000 X.25 to SNA Gateway or 2) SNA/X.25 Protocol Conversion.

If access to SNA is desired in order to communicate between HP and IBM computers, the options are 1)) an SNA gateway to IBM or 2) standalone SNA access from HP 3000 to IBM.

HP AdvanceNet for the Business Office

Company-wide Access Solution

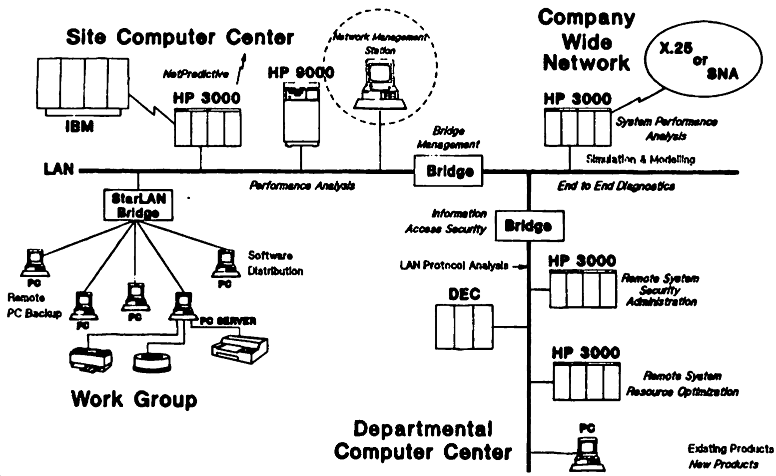
- A scalable, cost effective PC to mini integration
- HP STARLAN - the office network:
  - Use the regular phone wiring
  - Ease installation, add, move and change
- A complete range of office software



- Access to headquarters and other sites

The objective of Hewlett-Packard's Network Management is to provide network customers with the tools to create and manage data networks. Thus HP openview Network Management refers to a comprehensive family of network management tools and services. Many of the HP openview products are integrated under a common user interface, HP OpenView Windows. OpenView is mostly based on OSI standards, thus facilitating multi-vendor communication.

HP OpenView for Managing HP LAN's



HP ADVANCENET
BOS15

HEWLETT PACKARD

HP OpenView LAN Management Tools

HP OpenView LAN Bridge Manager

- **Bridge Management, Info Access Security, Traffic Monitoring**

HP Business Systems Plus

- **Remote PC Backup, Software Distribution**

HP OpenView Windows

HP OpenView Status & Diagnostic Monitor

HP OpenView Performance Monitor

HP OpenView Network Command Interpreter

HP AdvanceNet offers a comprehensive range of network support and services including planning and design, implementation, maintenance, and education that can be tailored to customer's unique requirements.

In summary, HP AdvanceNet for the business office solution provides the answer to the networking needs and requirements fo a general business environment.

Business Office

HP Multivendor Network Support

PLAN	IMPLEMENT	OPERATE
	Network Support	
Network Planning & Design	**Network Startup** **Network Prepare**	**NetAssure** **PPN Network Operation**
	Systems Support	

HP ADVANCENET
BOS18

HEWLETT PACKARD

Hewlett-Packard AdvanceNet for Regional Sales and Service

Alexander Henderson
Hewlett Packard

Many companies have facilities located over a wide geographical area. This may be for economic reasons (lower transportation costs, availability of materials), historic reasons (acquisitions), or competitive reasons (closer to the customer). These remote offices may vary widely in size, from a single salesperson on the road to a regional headquarters employing hundreds of people. The communications needs may also vary widely, from a simple link into a remote host to access a price file to networks designed to exchange and consolidate order information from many different branch offices.

The sales and service organization is a prime example of such a distributed organization. Sales and service can also be a key competitive advantage; excellent sales and service is one of the best ways for a company to distinguish itself. But this creates special problems. The cost of direct customer visits has skyrocketed in the last decade, yet sales and service remains the least automated function. Nevertheless, being responsive to the customer often means being able to provide increasing amounts of information (delivery commitments, order status, parts availability) on the spot.

The computer systems and networks used to provide that information are increasingly complex. Different applications reside on different vendors' systems; connecting those systems and maintaining the network is an increasingly difficult challenge. As companies seek to distinguish themselves by their sales and service, they also look to new applications that must be distributed over the entire network.

A good regional network must support communications options, allowing the users to mix and match choices in a way that minimizes costs while providing performance alternatives to offices of varying size. The network as a whole must also be flexible but cost effective.

The sales and service organization must have easy and quick access to information distributed throughout the company. It must be able to communicate that information both within the organization and to outside vendors and customers.

The systems and networks providing this information must be reliable and easily available from remote locations. Systems from different vendors must be able to exchange information. There should be services available to manage the network and to help created distributed applications.

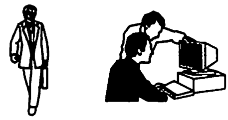
Networking Environment

Offices:

- **Geographically dispersed**
- **Varying sizes and communications needs**

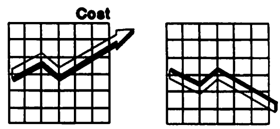
Sales & Service Organization:

- **Increasing cost of sales force and service organization**
- **Growing responsiveness to customer needs**



Systems:

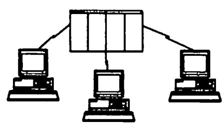
- **Increasing network maintenance complexity**
- **Sophisticated application development and version control**



Networking Needs

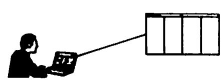
Offices:

- **Scalable branch/regional networking**
- **Cost-effective wide area network**



Sales & Service Organization:

- **Efficient information access and internal communication**
- **Ease of use/transparent access**

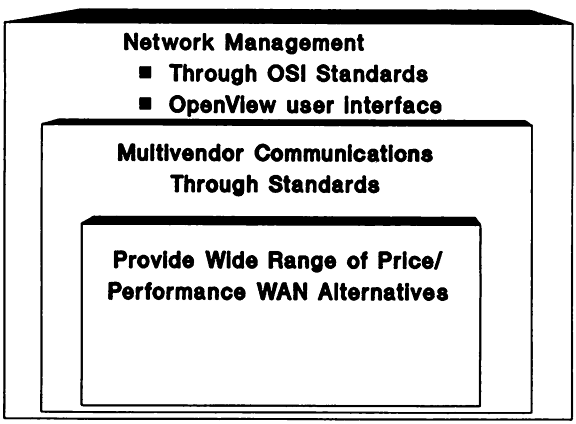


Systems:

- **High availability of network/systems**
- **Multivendor communications for company-wide access**
- **Distributed application services**

The Hewlett-Packard AdvanceNet Regional Sales and Service solution is the answer to these problems. Hewlett-Packard AdvanceNet is a strategy, not an architecture, developed to meet fundamental customer needs: communications between equipment of different vendors, focused solutions for specific business environments, and outstanding service and support. The Regional Sales and Service solution addresses the requirements of the geographically dispersed organization in three key ways. First, it offers a wide range of price/performance alternatives for wide area networking. Second, it offers networking based on Open Systems Interconnect (OSI) and de facto standards, allowing easy communication between differing vendors. Third, it offers network management based on the emerging OSI standards and Hewlett-Packard's friendly OpenView user interface.

Regional Sales and Service Network Strategy



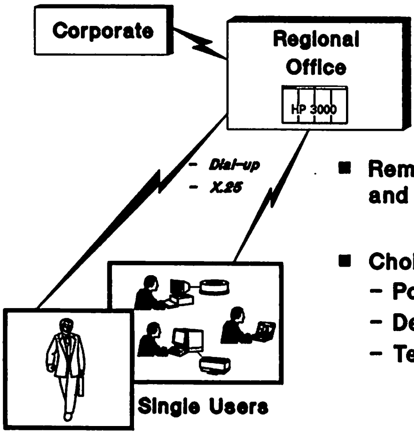
HP ADVANCENET
RSS06

HEWLETT PACKARD

The smallest office in the Regional Sales and Service solution is a single user. This will usually be an individual salesperson, either on the road or working at a remote location. This person is usually seeking remote access to information and internal communication (electronic mail). AdvanceNet offers connections for a variety of workstations, including portable PCs, desktop PCs, and terminals. Because of the low volume of information being exchanged, a simple point-to-point dial-up connection is usually the most cost-effective. When links to many different locations are important, a connection to a public X.25 network can be used. Tools such as AdvanceMail and Information Access give the user easy access to electronic mail and information in databases on the host system.

Regional Sales & Service

Individual Sales & Service Solution



- **Remote access to information and internal communication**
- **Choice of workstation:**
  - **Portable PC (286/386)**
  - **Desktop PC (286/386)**
  - **Terminals**

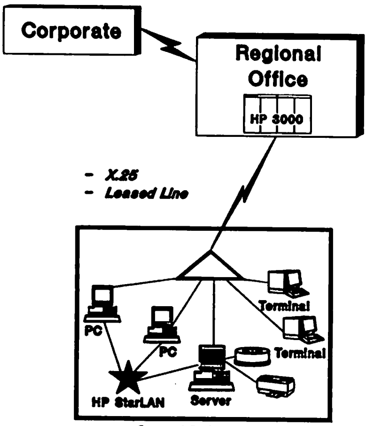
HP ADVANCENET
RSS08

HEWLETT PACKARD

In a medium-sized branch office, several workers need to both share work locally and access remote locations. A personal computer network provides the branch office with local file and peripheral sharing. The Hewlett-Packard 2334A PAD/Multiplexer provides remote connectivity, with terminal access to regional office computing power, applications and information. A variety of communications links are available. For low-volume traffic and infrequent use, a dial-up line is still the most cost-effective. Where there is medium to heavy traffic between offices, leased lines or X.25 networks make more sense. X.25 networks are the best answer when multiple other branch offices must be accessed.

Regional Sales & Service

Medium Branch Office Solution



2 - 16 Users

- **Remote access to information through PAD / Multiplexer**
- **Local access to information within the workgroup**
  - **PC server for sharing peripherals and data**

HP ADVANCENET
RSS09

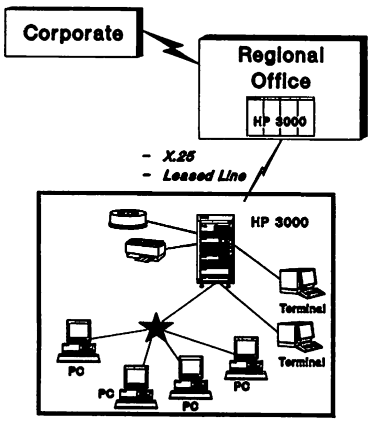
HEWLETT PACKARD

In a large branch office or regional headquarters, workers still need to both share work locally and access remote locations. But the applications or number of people now require the local data processing power of an HP 3000. Both terminals and PCs may be connected directly to the HP 3000, or the PCs may be hooked up via a LAN.

Other remote locations can be accessed with the Asynchronous Serial Link for low-cost access for low-volume traffic, with the NS Point-to-Point Link for high-performance access, or with the NS X.25 link, which provides cost-effective access to multiple regional and branch offices.

Regional Sales & Service

Large Branch Office Solution



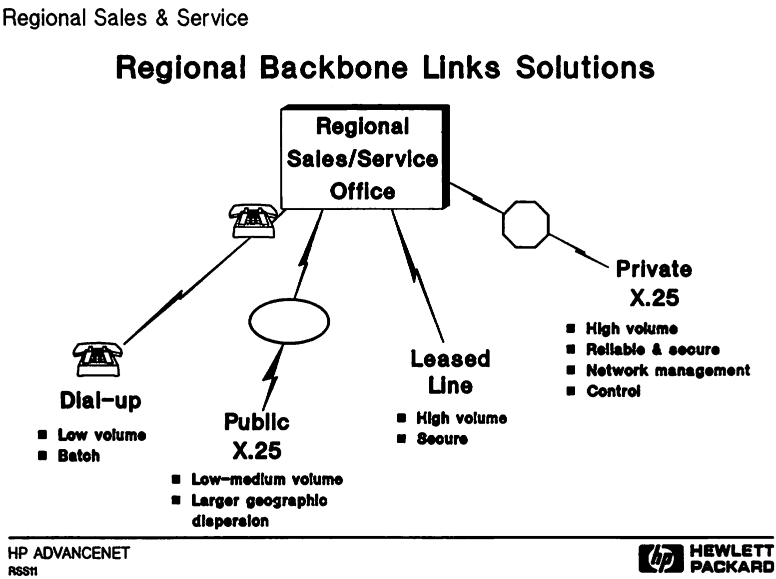
12 Users and Up

- Local & remote access to information
  - Low cost: Asynchronous serial network link
  - High performance: X.25 or point-to-point link

HP ADVANCENET
RSS10

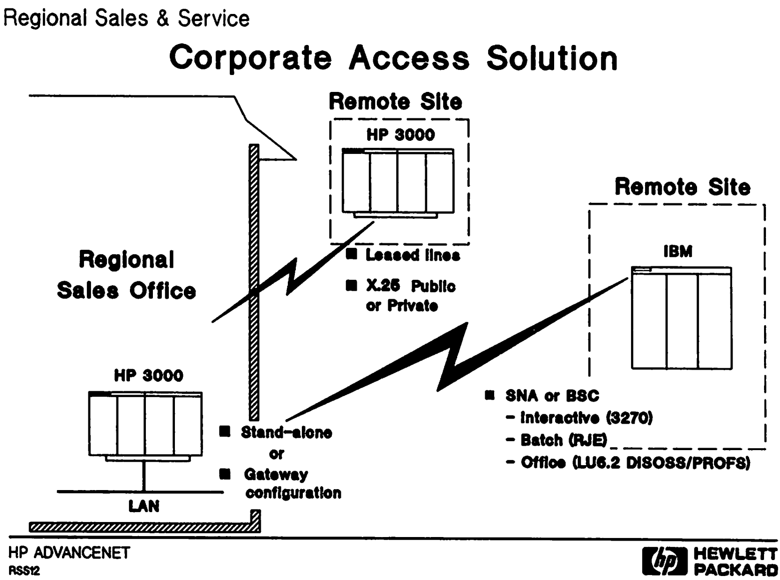
HEWLETT PACKARD

Thus within the regional network, AdvanceNet offers solutions on the full range of terminals, PCs, and larger systems. These solutions run over a variety of links, allowing a company to mix and match alternatives for the best price/performance solution for each office.



In the Regional Sales and Service environment there is often a need to share information with systems located in other remote data centers within the company (e.g., corporate headquarters, manufacturing plants, and engineering sites). These data exchange capabilities are need for end users who access remote applications and data bases, and for software processes running over multiple systems.

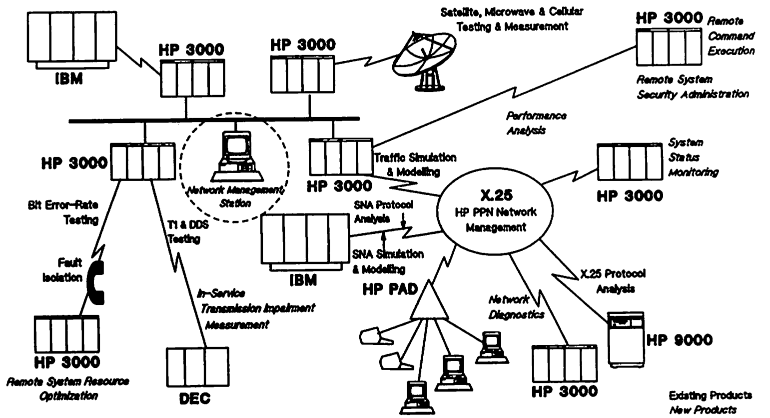
The way HP systems are interconnected with remote HP or IBM systems across a backbone network depends upon which backbone network is installed. For HP to HP communication, the X.25 based solutions already described provide the best peer-to-peer communications. For HP to IBM communications, AdvanceNet supports links over both SNA and Bisync. IBM services supported over these links include interactive services (3270), batch remote job entry (RJE), and electronic mail connections to DISOSS (over LU 6.2) and PROFS.



Network management is the ability to monitor, diagnose and control each component of a network. AdvanceNet now includes OpenView, Hewlett-Packard's comprehensive family of network management tools and services. Integrating many of the OpenView products is OpenView Windows, a graphical interface offering the network manager a visual network map and common simple "look and feel" to each tool. Each of the other OpenView products offers a specific tool for a given network management task. Most of the OpenView products are based on the OSI standards, allowing the network manager to ultimately manage a network incorporating products from many different vendors.

Regional Sales & Service

HP OpenView for Managing HP WAN's



HP ADVANCENET

RSS15

HEWLETT PACKARD

Regional Sales & Service

HP OpenView WAN Management Tools

HP OpenView Windows

- **Integrated User Interface & Application Integration Point**

HP OpenView Status & Diagnostic Monitor

- **Remote System Status Monitoring, Network Diagnostics, Fault Isolation**

HP OpenView Performance Monitor

- **Remote System Performance Analysis**

HP OpenView Network Command Interpreter

- **Remote Command Execution**
- **Remote System Resource Optimization & Performance Analysis with OPT/3000**
- **Remote System Security Administration with Security Monitor/V**

HP OpenView ITIMS Manager

- **Centralized Transmission Impairment Measurement through HP OpenView Windows**

HP ADVANCENET
RSS15B

HEWLETT PACKARD

The Regional Sales and Service solution also includes services to support distributed applications. This includes NetDelivery, an asynchronous store and forward facility, and X.400, the OSI message handling system.

Complex, geographically dispersed networks need strong service and support. Hewlett-Packard has always been known for its service, and AdvanceNet offers a full range of network support services, including network planning and design, start-up and implementation, maintenance, and education. Any and all of these services can be tailored to meet a customer's unique requirements.

In summary, the AdvanceNet Regional Sales and Service solution offers the products and services needed to build a strong, cost-effective wide area network.

Regional Sales & Service

HP Multivendor Network Support

PLAN	IMPLEMENT	OPERATE
	Network Support	
Network Planning & Design	Network Startup Network Prepare	NetAssure PPN Network Operation
	Systems Support	

HP ADVANCENET
RSS18

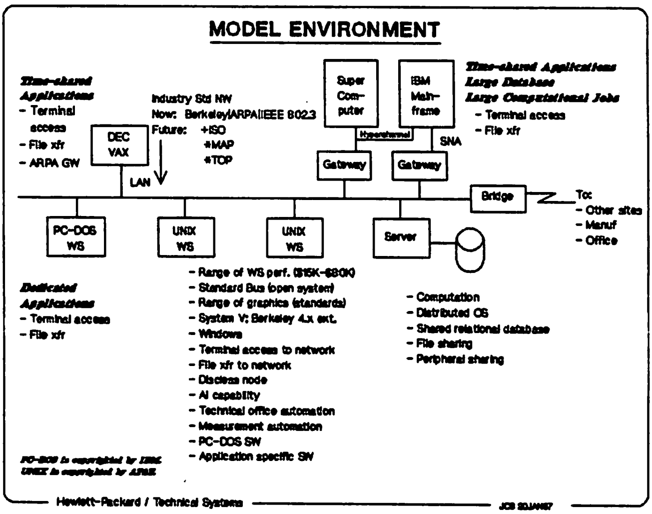
HEWLETT PACKARD

HP AdvanceNet for Engineering
Dave Morse

Hewlett-Packard Company
3404 East Harmony Road
Fort Collins, CO 80525

INTRODUCTION

As one of the five solutions in the HP AdvanceNet offering, HP AdvanceNet for Engineering addresses the networking needs of technical professionals engaged in engineering and other technical pursuits. This solution features the same emphasis on standards common to the other solutions. The solution is best understood by considering a model computing environment for engineering.

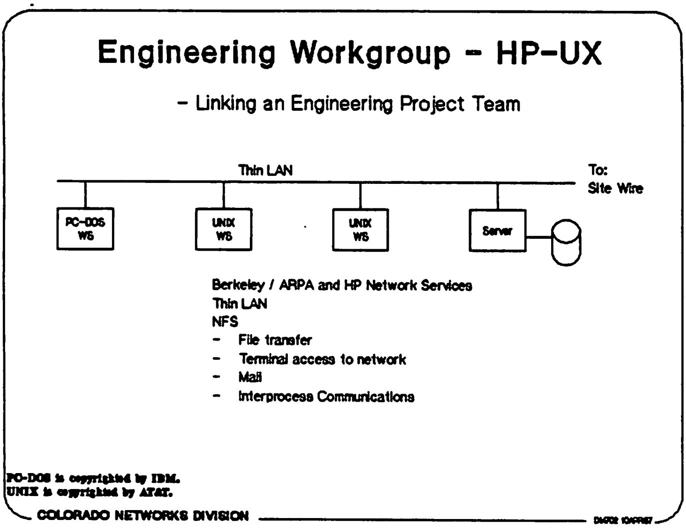


The diagram of the environment shows many of the key characteristics of both the computers and the network. A major trend in the engineering area in the past few years has been a move to engineering workstations and acceptance of the UNIX operating system as a defacto standard. These workstations offer many advantages in terms of powerful graphics and consistent performance; but in order to be effective, they must easily integrate with the installed base of timeshare computers and other larger computers which may be added in the future. The resulting environment represents a range of computing power from personal computers to mainframes and super computers. In almost all cases, these computers will be supplied by several different vendors. In order for users to realize the maximum benefit of this environment, they should retain the desirable characteristics of the timeshare environment - easy information sharing and centralized system management - and also gain the benefits of the workstations in terms of distributed computing power. The network plays the key role in providing this.

The basic purpose of the network is to provide information and resource sharing. Users should be able to transfer files from one computer to another, log on to other computers, run applications on other computers, run applications on a local computer using data on remote computers, access peripherals connected to any computer and, in general, make the best use of the available resources to perform a wide variety of different tasks. In fact, it is not possible to do an effective job of providing computing for engineers without providing the supporting network.

The Engineering Solution, like the other HP AdvanceNet solutions, is comprised of modules. There are five modules in the Engineering Solution: Engineering Workgroup, Engineering Computer Center, Site Computer Center Access, Site Wire, and Company-wide Access. Each module consists of a collection of products which together meet the user requirements.

The first three modules represent a three-tiered hierarchy commonly found in engineering environments - workstations, super minicomputers, and mainframes. The workstations and super-minicomputers are often administered by the engineering department. The mainframes are often facility resources administered by the EDP or MIS departments.

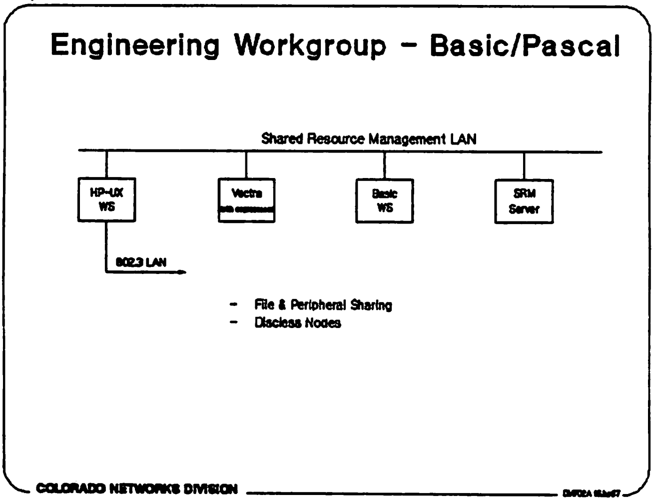


The most effective and productive way to connect a group of workstations is with a local area network (LAN). The LAN that has emerged as a standard for engineering networks is the IEEE 802.3 LAN. Early engineering networks utilized the Ethernet protocol, which served as basis for the IEEE 802.3 standard. HP offers IEEE 802.3 LAN as the basis for the engineering workgroup of UNIX workstations. For compatibility with existing networks, the Ethernet protocol is also supported. The IEEE 802.3 standard only defines part of the protocols necessary to provide communication among the computers. The other protocols employed are the Transmission Control Protocol (TCP) and Internet Protocol (IP) and the Berkeley and ARPA network services. The IEEE 802.3 standard defines the link used to connect the computers, TCP/IP provides a reliable connection from one computer to another, and the Berkeley and ARPA network services provide the specific functions required, such as file transfer, virtual terminal, etc.

The IEEE 802.3 standard allows for two types of cable - thin and thick. Because of ease of installation and configuration, HP recommends use of the thin cable for the engineering workgroup.

One necessary capability not provided by either Berkeley or ARPA services is the ability to share files without copying the entire file from one computer to another. HP has augmented the Berkeley and ARPA services with an HP developed service called Remote File Access. Recently, a service known as the Network File System* (NFS) has been endorsed by a number of vendors and has emerged as a defacto standard for file sharing. One of the major advantages of NFS is that it is independent of the operating system and thus allows sharing of files among computers with UNIX and other operating systems. HP has announced NFS for the HP 9000 computers with initial shipments planned for late 1987.

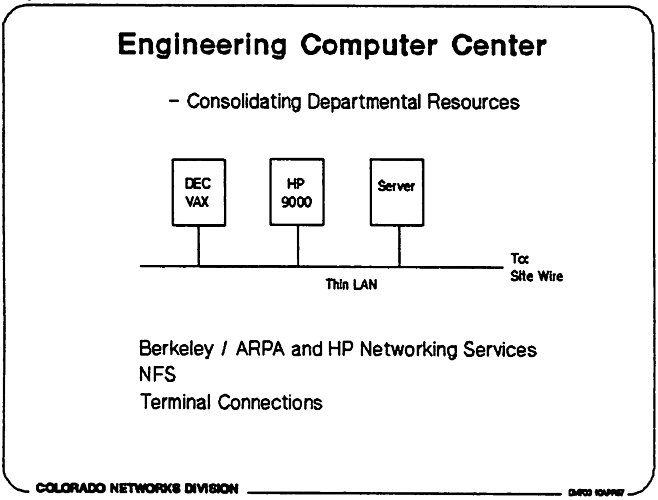
* NFS is a a trademark of SUN MicroSystems



Many engineering applications require use of computers to control various types of test and measurement equipment. HP offers several computers optimized for this task. One widely used computer is an HP 9000 Series 200 or 300 running the BASIC operating system (Rocky Mountain BASIC). The HP network for these systems is the Shared Resource Manager (SRM). The SRM features peripheral and file sharing and allows operation of the workstations without local discs.

SRM also supports the PASCAL workstation and HP-UX. SRM networks can be connected to IEEE 802.3 LANs through a workstation running HP-UX or a Vectra PC with the BASIC co-processor.

ENGINEERING COMPUTER CENTER MODULE



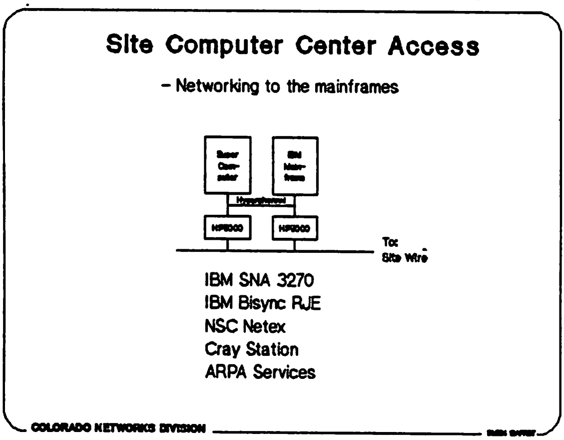
The engineering computer center represents the second tier in the engineering computing hierarchy. Computers in the engineering computer center are typically departmental resources, shared by several project teams. A timeshare super-minicomputer, such as the Digital Equipment Corporation VAX is very commonly used in this environment. Recently introduced HP Precision Architecture computers such and the HP 9000 Models 825, 840, and 850 will be installed here.

The engineering computer center could also house various types of servers for the engineering workgroups. These servers could manage large ensembles of discs or other peripherals such as laser printers. An advantage of putting servers in the engineering computer center is that they are centralized with the timeshare computers for convenient disc backups. Placing the majority of the discs and other peripherals here also isolates the workgroups from the noise generated by these devices.

HP's recommended wiring for the engineering computer center is again the ThinLan cabling. This allows for easy configuration of the computer center and permits convenient reconfiguration as necessary. The ThinLan network for the computer center can then be connected to the site backbone for communication with other workgroups or computer centers.

The ARPA and Berkeley network services can be used with any computers running UNIX. For example, the HP 9000 Series 800 computers can all be configured with ARPA and Berkeley network services to augment the HP-UX operating system. DEC VAX computers running the VMS operating system can be equipped with ARPA services via software packages available from DEC and several third parties. An alternative means to connect DEC VAX computers with VMS is to install HP Network Services on the VAX computer. This product provides HP AdvanceNet Network File Transfer (NFT), allowing file transfers between the VAX and any HP computer supporting NFT. The HP Network Services for the VAX run in user space and utilize standard DEC LAN hardware, permitting coexistence with DECNet. The engineering computer center would also provide terminal connections for the various timeshare computers.

SITE COMPUTER ACCESS MODULE



The site computer center is the province of the mainframe and supercomputer. IBM and IBM compatible mainframes are commonly found here. Engineers utilize these resources to execute jobs requiring extensive computational power or to access large databases. Because of the dominance of IBM in this environment, required networking capabilities fall into two categories - IBM communications and "other".

Today the most commonly used protocol to communicate with IBM mainframes is IBM Systems Network Architecture (SNA). An older protocol, Binary Synchronous Communications (Bisync), is still in use in some installations. Either of these alternatives offers convenient communications to IBM because the engineering computers emulate standard IBM devices, such as interactive terminals or remote job entry stations. From the IBM mainframe's perspective, it is communicating with another IBM device. This greatly simplifies the task of the mainframe system managers, since they deal with standard IBM software. HP offers both SNA and Bisync communications products for communication with IBM mainframes.

A disadvantage of utilizing standard IBM SNA communications is that the performance is generally limited to that attainable over 56 Kbit/second links, far short of what can be obtained with a LAN. Because of the performance limitations, many site computer centers support alternative, non-IBM, connections to the mainframes.

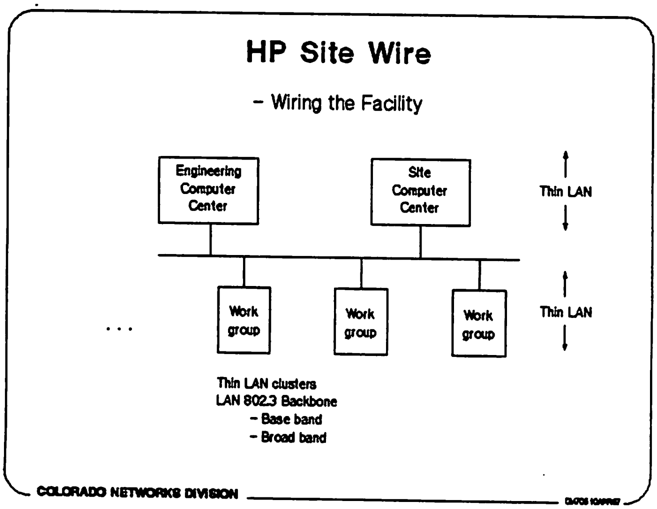
Probably the most commonly encountered product is Hyperchannel, provided by Network Systems Corporation (NSC). Hyperchannel features a 50 Mbit/second link and supports a wide variety of computers in addition to IBM. Hyperchannel connections are available from NSC for HP 9000 computers.

It is also possible to support the ARPA services on an IBM mainframe. Products are available from a variety of vendors. IBM also sells a TCP/IP/Ethernet product. Many of these products are new on the market and are not commonly installed. Where they are supported by the site computer center, they offer an additional high speed connection from the HP computers to IBM.

Many site computer centers also contain supercomputers such as Crays. Cray computers running the Cray Operating System (COS) support access via a protocol called Cray Station, which runs over 50 MBit/second Hyperchannel hardware. Cray Station software is available from Cray Research for the HP 9000 computers. Cray computers running the Cray version of UNIX (UNICOS) support ARPA services over an Ethernet LAN and can communicate with HP 9000 computers using this protocol.

In general, communication with the site computer center will be via gateways between the engineering or facility LAN and the computer center. The gateways provide access to the computer center for other computers on the network and eliminate the need to install individual mainframe communication links for each computer. HP 9000 Series 300 computers can serve as these gateways.

HP SITEWIRE MODULE



The network of choice for most engineering applications today is IEEE 802.3. IEEE 802.3 supports two cabling options ThinLAN and ThickLAN. HP's recommended wiring scheme utilizes ThinLAN clusters for the engineering workgroups and engineering computer center. These ThinLAN clusters are connected to a ThickLAN backbone which runs throughout the facility. A device known as a ThinLAN Hub provides the connection between up to 4 ThinLAN subnets and the ThickLAN backbone. The ThinLAN and ThickLAN segments run at the same 10 Mbit/second link speed.

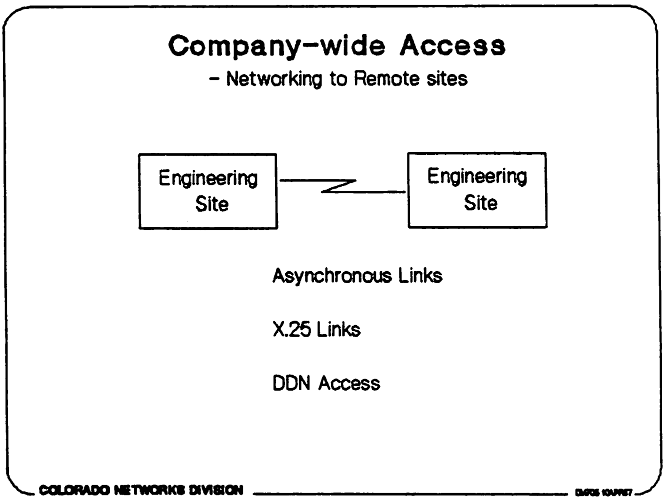
Small engineering networks can be created by using only a single ThinLAN network or by connecting up to four ThinLAN clusters with a single ThinLAN Hub.

ThickLANs can serve as backbones for networks of 1-2 kilometers in length. ThickLANs are baseband networks.

Broadband backbones are utilized to connect larger facilities or campuses. Broadband backbones use cable television technology to cover distances spanning many kilometers. Broadband backbones have the additional advantage of supporting many channels of communication. A single LAN can thus be used for computer to computer communications, terminal to computer communications, closed circuit television, and a variety of other uses. For this reasons, broadband backbones are sometimes installed instead of baseband backbones even for small networks.

HP supplies baseband networks. HP also supports broadband backbones through the use of recommended products from Ungermann Bass, such as the Buffered Repeater, which connects ThinLAN clusters to broadband backbones.

COMPANY-WIDE ACCESS MODULE



Although many engineering networks involve only a single site, there is often a requirement to connect engineering communities at several locations into a common network.

If the traffic between sites is not extensive, the simplest alternative is to use dial-up telephone lines and asynchronous modems. HP 9000 computers support standard UNIX communications services such as uucp and mail which utilize these asynchronous modems.

In addition, the UNIX communications services can utilize X.25 networks through the use of an HP supplied X.25 multiplexer. The X.25 networks have the additional benefit of more reliable data transmission. In many cases they are also more cost effective than dial-up communications lines.

X.25 communications can be provided by public X.25 networks such and Telenet in the United States or Transpac in France. HP also provides switches which can be used to create a private X.25 network which would carry traffic for only a single company. Such a network may be of interest, for example, if there are special security or performance requirements. A private X.25 network also allows very tight control of network operations.

SUMMARY

HP AdvanceNet for Engineering provides comprehensive, standards based networking which meets the diverse needs of today's engineering community. Since HP AdvanceNet is based on standards, it provides a network which will evolve and endure for many years. Since it supports a multivendor computing environment, it offers flexibility in the selection of engineering computers.

HP is an active participant in the organizations defining future networking standards, such as the IEEE 802 committee, X/OPEN, the MAP Users Group, and the Corporation for Open Systems (COS). HP chairs several key working groups in these organizations.

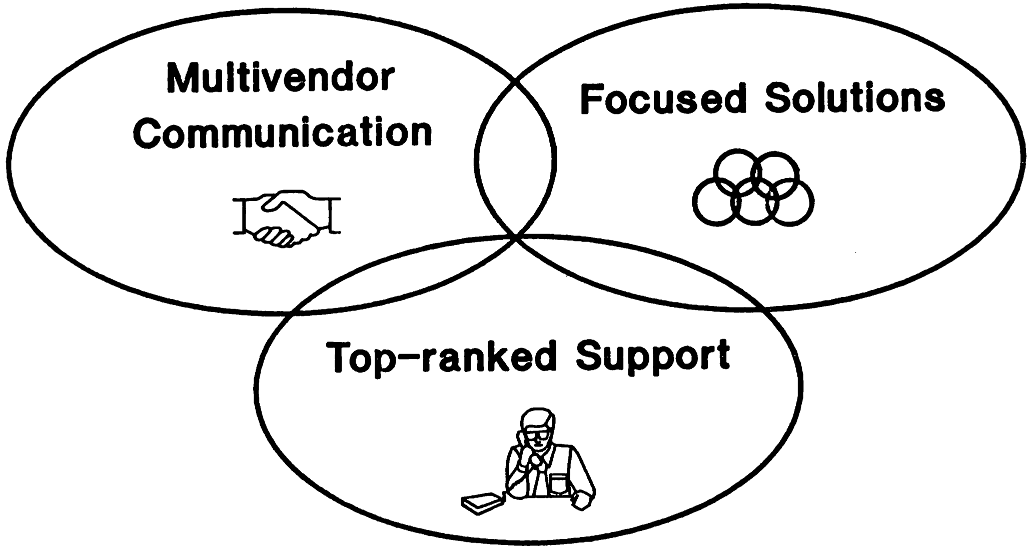
As the requirements for engineering networks grow, HP AdvanceNet for Engineering will grow with them.

k
/87

2030-11

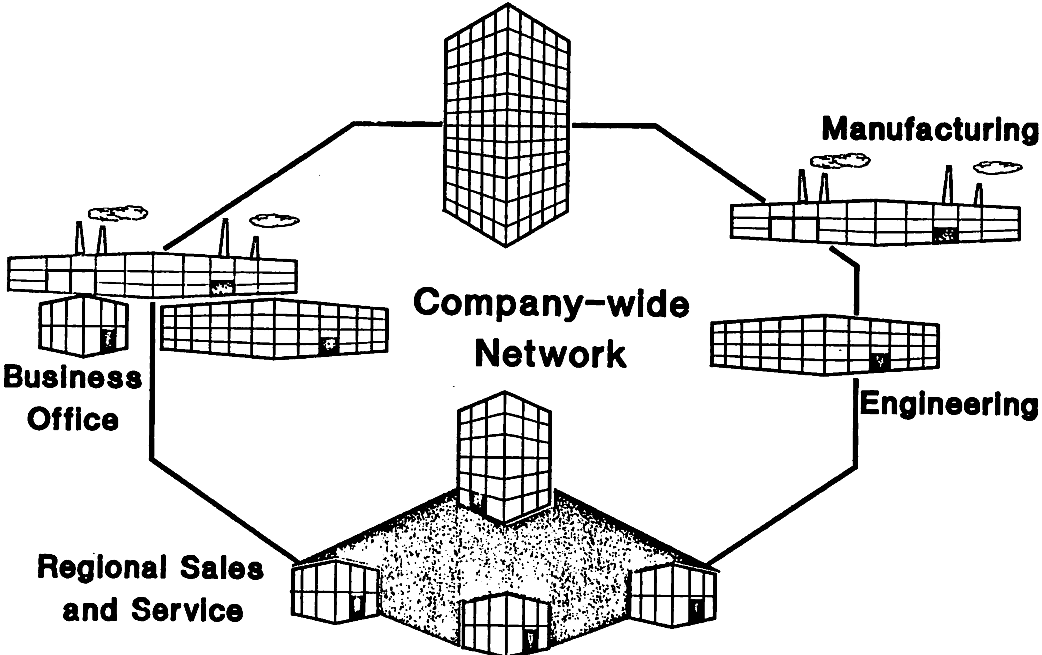
HP AdvanceNet

A *Strategy* for Integrated Networked Solutions



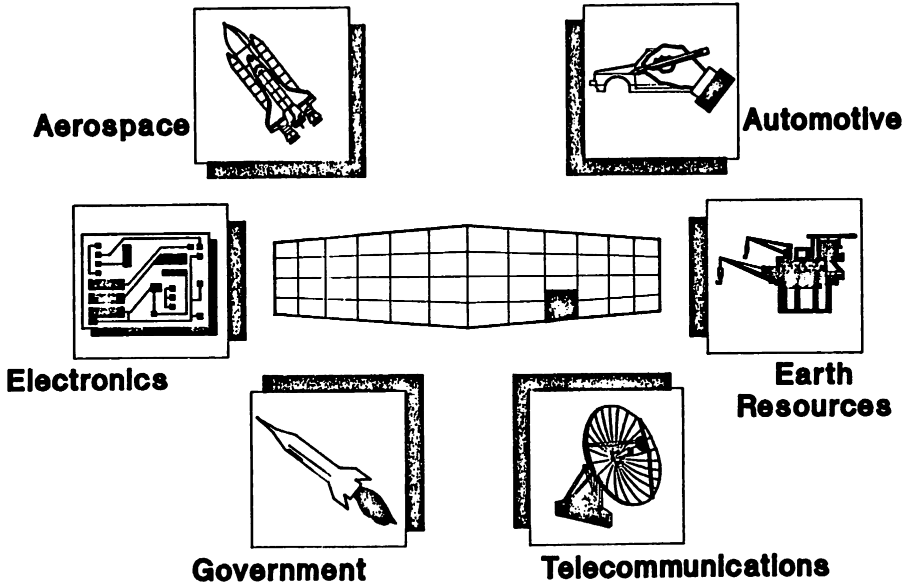
2030-12

HP AdvanceNet: 5 Networking Solutions

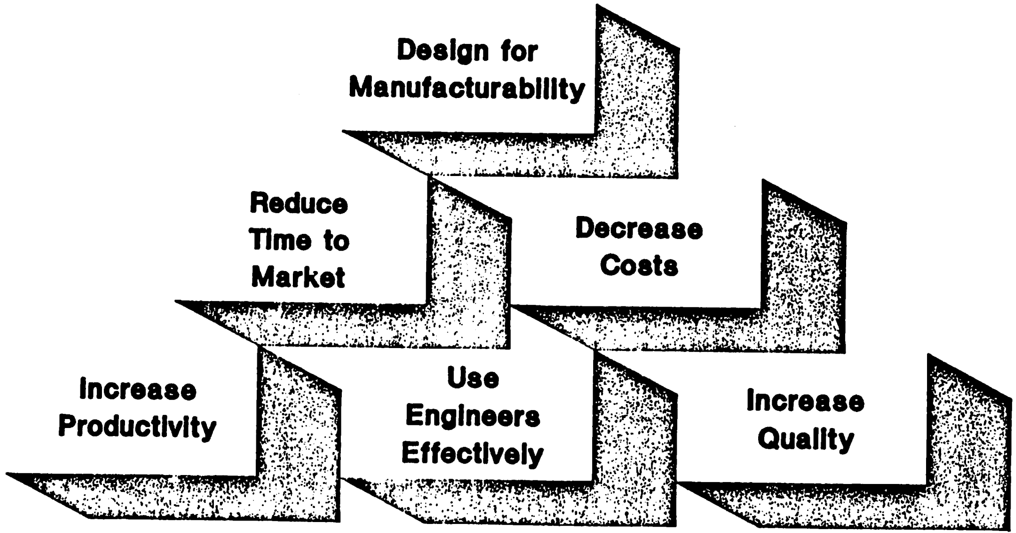


HEWLETT PACKARD

HP AdvanceNet in Engineering



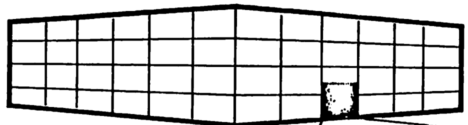
Engineering Management Challenges



Engineering Environment

- Frequent adds, moves and changes
- Diverse equipment from many vendors
- Mixture of offices/open space
- Office tasks and engineering tasks

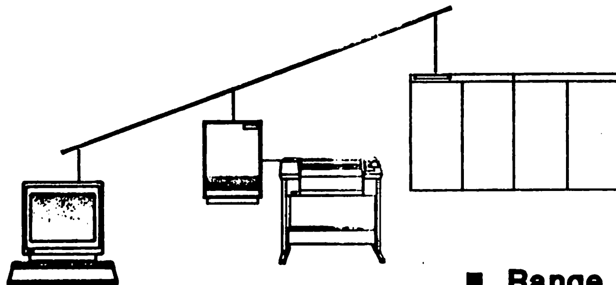
Trends In Engineering



- Engineering/CIM integration
- More simulation; less build and test
- Engineering resources scarcer and more expensive
- Tools more effective and less expensive

2030-17

Engineering Computing Environment

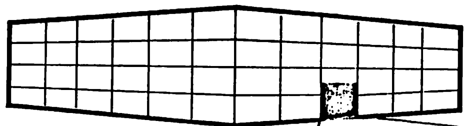


- Range of computing power
- Multivendor networks
- Multiple versions of information
- Application dependent data
- System dependent applications

2030-18

Engineering

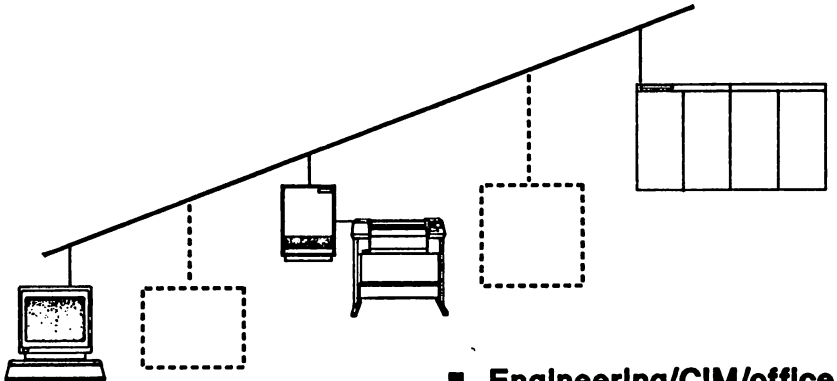
Engineering Computing Trends



- Higher speed systems and networking
- Workstation/server integration
- Emergence of protocol standards
- Proposals for management standards

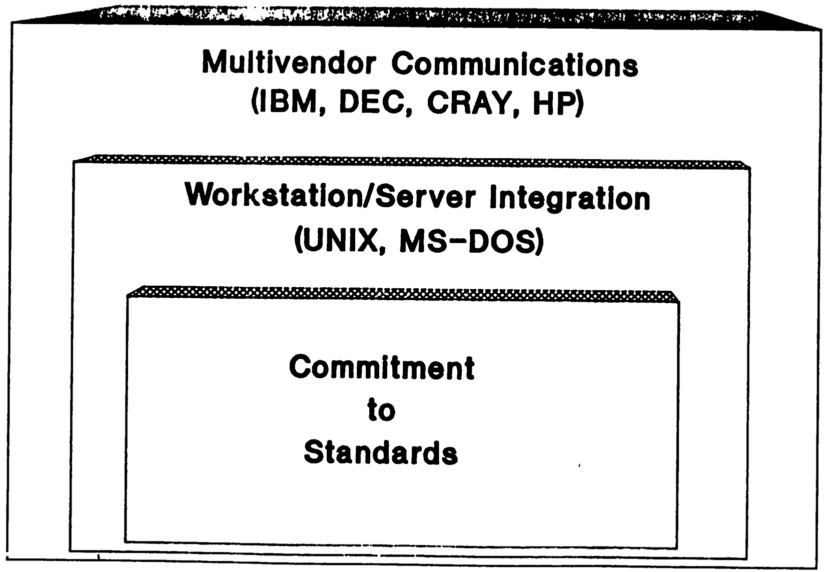
2030-19

Engineering Computing Needs



- Engineering/CIM/office integration
- Invisible networks
- Simple, standard human interface
- Management control

Engineering Network Strategy



2030-21

HP and Engineering Network Standards

OSI Architecture

- Phase 1 – ARPA/Berkeley
- Phase 2 – ISO

Multivendor Testing Commitment

- Conformance
- Interoperability

Hewlett-Packard Standards Commitment

Leadership in Link Technology

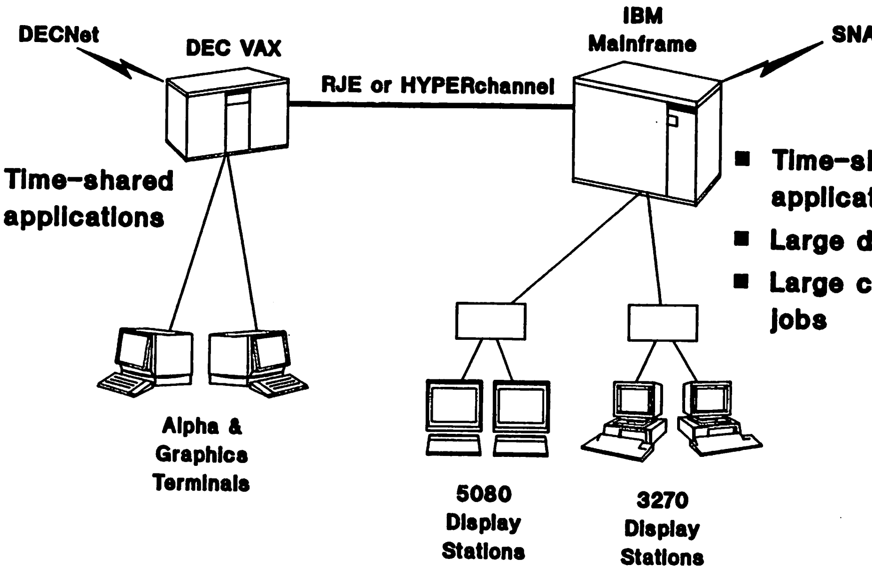
- StarLAN 10
- ThinLAN
- HP X.25 Private Packet Network

43 Committees/ Task Groups

- Chair of 10 BASE T
- Chair of IEEE 802.3
- Founder of Network Mgmt. Committee
- Founding member of COS

2030-22

Time-share Environment

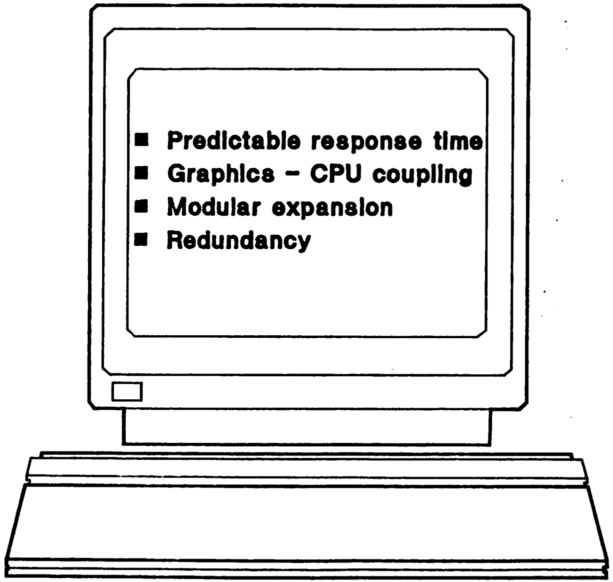


- Time-shared applications

- Time-shared applications
- Large database
- Large computational jobs

2030-23

Engineering Workstations

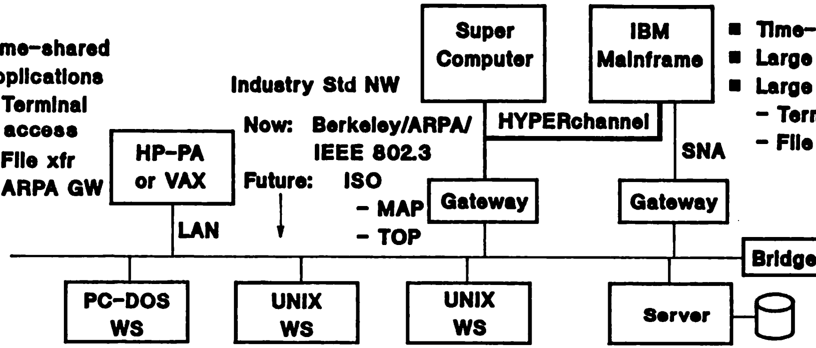


2030-24

Engineering

Model Environment

- Time-shared applications
  - Terminal access
  - File xfr
  - ARPA GW



- Time-shared applications
- Large database
- Large computational jobs
  - Terminal access
  - File xfr

To:
- Other site
- Mfg.
- Office

- Dedicated applications
  - Terminal access
  - File xfr

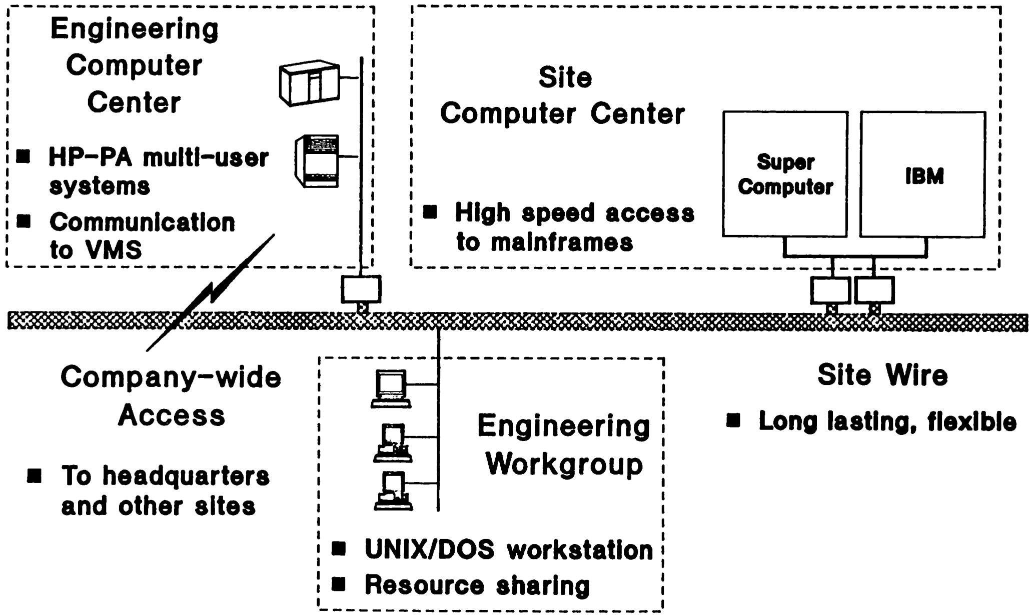
- Range of WS perf. ($5K - $80K)
- Standard Bus (open system)
- Range of graphics (standards)
- System V; Berkeley 4.X ext.
- Windows
- Terminal access to network
- File xfr to network
- Discless node
- AI capability
- Technical office automation
- Measurement automation
- PC-DOS SW
- Application specific SW

- Computation
- Distributed OS
- Shared relational database
- File sharing
- Peripheral sharing

ENG11d

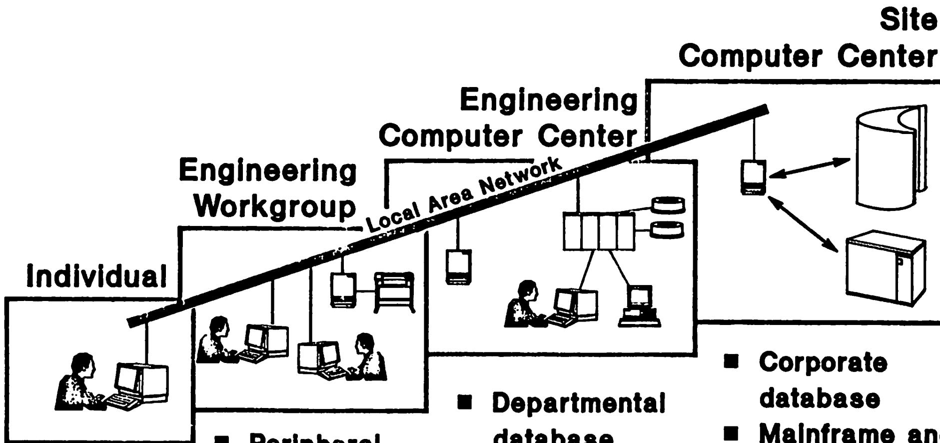
2030-25

HP AdvanceNet Solution



2030-26

Engineering Computing Tiers



Individual
- Local files
- Workstation computing

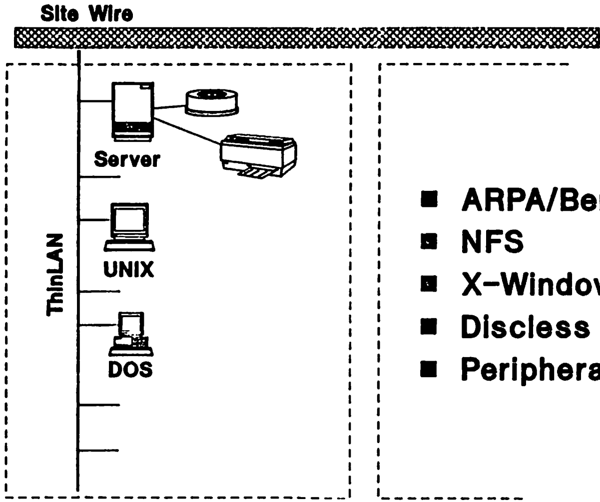
Engineering Workgroup
- Peripheral sharing
- File Sharing

Engineering Computer Center
- Departmental database
- Departmental computing

Site Computer Center
- Corporate database
- Mainframe and super computing

Engineering Workgroup Solution

Linking an Engineering Project Team



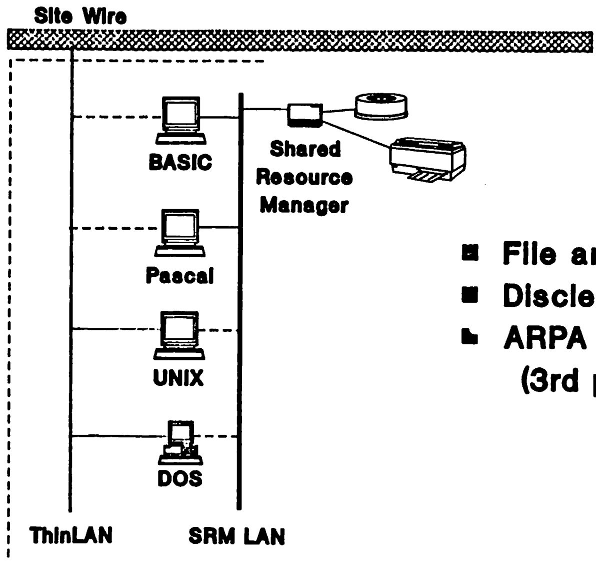
Workgroup Project #1

Workgroup Project #2

- ARPA/Berkeley/HP NS (TCP/IP)
- NFS
- X-Windows
- Discless nodes
- Peripheral and file servers

Engineering Workgroup Solution

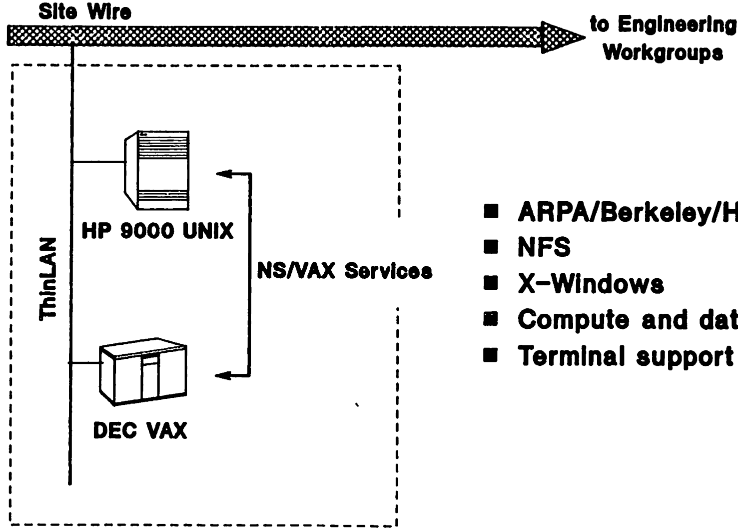
BASIC/Pascal Workstations



- File and peripheral sharing
- Discless nodes
- ARPA support via FUSION (3rd party)

Engineering Computer Center Solution

Consolidate Departmental Resources



- ARPA/Berkeley/HP NS (TCP/IP)
- NFS
- X-Windows
- Compute and database servers
- Terminal support

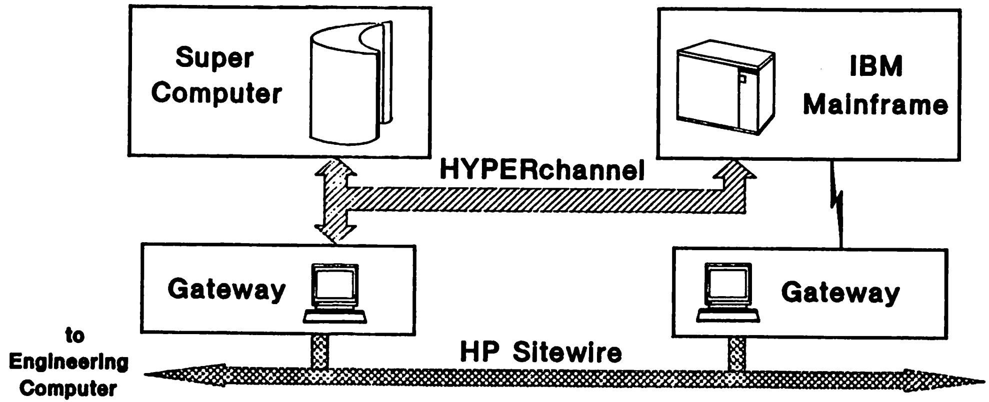
HEWLETT PACKARD

2030-30

Engineering

Site Computer Center Access Solution

Networking to the Mainframes

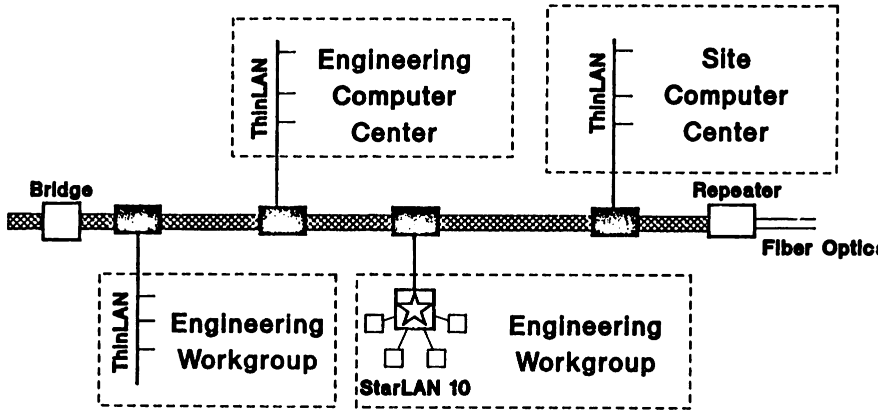


- NSC Netex
- Cray Station
- ARPA Services

- IBM SNA 3270/3770
- IBM BSC 2780/3780
- IBM Channel Attach
- IBM 5080 Emulation

Site Wire Solution

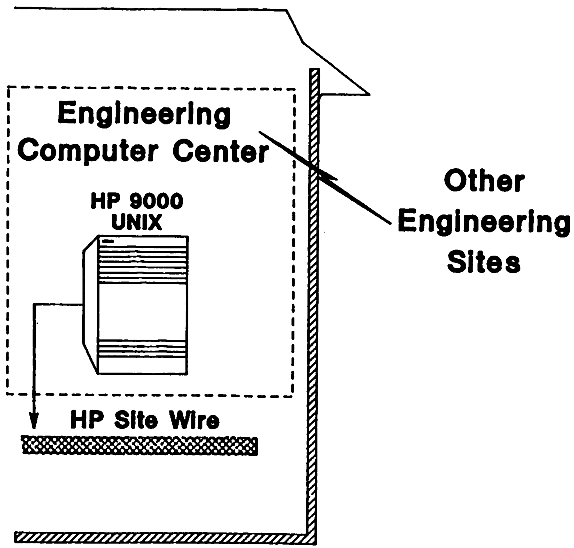
Wiring The Facility



- **Backbone**
  - Ethernet/IEEE 802.3 ThickLAN
  - Baseband or broadband
- **Subnets**
  - ThinLAN and StarLAN 10

Company-wide Access Solution

Networking to Remote Sites



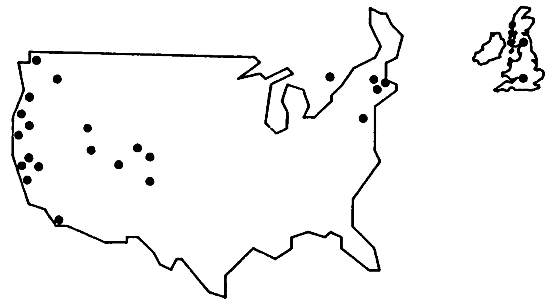
- X.25 links
  - ARPA services
  - Level III access
- DDN access
  - ARPA/TCP/iP
- Asynchronous links
  - uucp, cu

2030-33

Engineering

HP Internet: Connecting HP R&D Labs

Largest TCP/IP Based Network in the World



Sites

- 4500 nodes
- 27 divisions
- Expansion to include Canada, Europe, Japan, Singapore, Australia

Operating Systems

- HP-UX
- VMS
- TOPS 20
- Apollo Domain
- Symbolics LISP
- MS-DOS

Technology

- Broadband
- Baseband - (ThickLAN, ThinLAN, StarLAN)
- Satellite links
- Serial links: 9.6 Kbps, 56 Kbps, T-1, X.25

2030-34

Engineering

HP Multivendor Network Support

PLAN	IMPLEMENT	OPERATE
	Network Support	
Network Planning & Design	Network Startup Network Prepare	NetAssure PPN Network Operation
	Systems Support	

2030-35

HP AdvanceNet for Engineering

HEWLETT PACKARD

THE CIM SOLUTION

BRICE L. CLARK
HEWLETT-PACKARD
ROSEVILLE, CALIFORNIA

INTRODUCTION

In today's business environment, failure to respond to changing and tougher competition for world markets can mean the permanent failure of a business or even an entire industry. Corporations are increasingly turning to their manufacturing managers to improve profits, product quality and costs to help stay competitive.

These objectives lead manufacturing managers to seek solutions with quality programs, productivity improvements and flexible manufacturing. Perceptive managers realize that information is the key to making these changes, and that CIM (computer integrated manufacturing) is the key to the productive utilization of information resources.

The benefits of CIM can be summarized as follows:

-- higher quality -- CIM means reduced scrap and rework, lower warranty costs and higher customer satisfaction.

-- increased productivity -- With improved resource utilization, reduced cycle times and lower overhead, productivity improves.

-- greater flexibility -- CIM translates into lower inventories, faster response to market demand and less time to get products to market.

In short, CIM means lower manufacturing costs, higher profit margins and better positioning for growth -- a healthier bottom line.

Among the specific applications that deliver these benefits are:

quality -- Quality Decision Management (QDM) from HP;

(RQM) from

Automated Technology Associates.

productivity -- HP's Maintenance Management Package (MMP);

Materials Management (MM) and Production

Management (PM)

Dispatcher from Logisticon.

flexibility -- MM and PM from HP;

Monitrol from Hillco;

StarNet from Denniston and Denniston;

AIM from Billes and Associates;

HP's Just-In-Time software.

As today's manufacturers develop and implement their CIM plans, computer networks, and the information they manage and deliver, are increasingly important.

This paper provides an overview of Hewlett-Packard's CIM Networking Solution. The CIM solution is based on HP AdvanceNet, HP's overall networking strategy, which delivers flexible, scalable solutions and embodies a strong commitment to standards-based networking. Any successful CIM implementation requires an efficient information network as the communications foundation upon which the solution is based.

CIM ACTIVITIES AND ENVIRONMENTS

A discussion of networks begins with a look at basic business and environmental factors. The primary activities that go on in businesses today include administration, planning and control; manufacturing; marketing and sales; engineering; and facilities administration. The activities performed in a given area establish the information needs of that area, and information needs help determine the most appropriate networking technology.

Since networks operate over physical media (at least within buildings), the physical environment will also have an important effect on the choice of networking technology. Office settings tend to be clean, quiet, and include phones on every desk. Computer centers are custom environments designed for computers from the outset, while factory production areas are typically noisy, dirty and can be quite large. Unlike the office, production areas tend to have few phones.

ISLANDS OF AUTOMATION

One of the key problems manufacturing managers face today are the "islands of automation" that exist on their factory floors. These islands are the result of the efforts of many manufacturers to streamline different parts of the manufacturing process. Frequently, each automated process has been designed and engineered without much thought of integrating it with other processes.

Many companies are starting to see the benefits of integrating their information sources, but are having problems communicating with a wide range of "automation islands." Advanced manufacturing companies are looking to information automation as the key to moving beyond "islands of automation" toward an integrated automation environment.

Networks can often make important contributions to control and integration of the physical automation process. Islands of automation merge as individual processes are refined to match the needs of upstream and downstream processes. This merging creates the need for "real-time" communication from machine-to-machine or control point-to-control point.

As the whole process becomes further automated, a network enables information to flow fast enough to permit real-time corrections that can either prevent errors before they happen or spot them fast enough to correct them at minimal cost.

Unfortunately, the manager who recognizes the potential of information-handling networks often encounters a horrible sight just beyond the horizon. At HP, we call this the "CIM Barrier."

THE CIM BARRIER

It's sad but true: the manufacturing manager who wants to improve effectiveness through CIM is often held back by the limitations of poor or non-existent wiring systems and the lack of multivendor compatibility that have characterized the past.

There are several key problems with the traditional approach to using computers in manufacturing. These are:

--sparse connectivity with point-to-point connections to large, inflexible systems in the computer center;

--slow information flow via paper;

--lack of communication among the computers of different vendors;

--low networking expertise in manufacturing companies;

--no systematic approach to wiring.

Collectively these problems make up the CIM Barrier. HP's CIM Networking Solution is designed to enable manufacturers to break through this barrier. The solution is divided into several modules, each of which addresses a particular communications problem. In addition, the HP AdvanceNet solution enables manufacturers to implement their networks in affordable, manageable stages.

CIM NETWORKING MODULES

The modules that comprise the CIM networking solution reflect the hierarchical structure typical of manufacturing sites. This structure starts at the level of shopfloor devices, and moves up to cell controllers, area managers, plant hosts and corporate hosts. Establishing effective communications among the levels of this hierarchy is addressed specifically in the Plant Area Management Module, but such communications can also be seen as the overall goal of CIM.

HP SiteWire Module

Within a manufacturing facility, two kinds of networks are generally found: the site backbone network and subnetworks. The backbone is a common communications channel that connects different workgroups throughout a facility. Subnetworks provide the specific functionalities needed in offices, engineering departments and production areas.

HP SiteWire is the name of the HP AdvanceNet communications wiring infrastructure. CIM networks requires a plant-wide communications backbone to connect people with information; subnetworks need easy access to the backbone from anywhere in the plant. In addition, the backbone network should be able to accommodate moves and changes easily.

The two HP SiteWire backbone options are based on industry standards. The primary and most versatile backbone option is based on the IEEE 802.7 broadband standard. It allows multiple information channels of voice, video and data. It also lets users mix terminals, point-to-point links, LANs and more on a single cable, and it supports an important new industry standard: the Manufacturing Automation Protocol (MAP).

IEEE 802.3 baseband is the second alternative, better suited to less complex situations that don't require video, voice or specialized data services.

End User Workgroup and Access Module

HP AdvanceNet offers a wide range of end-user solutions to enhance local departmental productivity while providing access to information throughout the plant. Options provide for multivendor terminal clusters and the latest in industry-standard LANs. Each option focuses on a specific area of the plant: planning and control, production engineering and the shop floor.

Thanks to terminal cluster solutions, a common problem of the past -- connecting terminals to the systems of different vendors -- isn't a problem of the present. Hewlett-Packard has entered into a special marketing agreement with Ungermann-Bass, the leading independent vendor of terminal servers, in order to provide this capability for its customers. The UB terminal servers operate over both broadband and baseband, and handle systems and terminals from a wide variety of vendors.

A planning and control staff can enjoy greater productivity and effectiveness using industry-standard StarLAN to connect PCs with information systems over low-cost twisted pair. HP StarLAN subnetworks can connect to either CIM backbone; to allow for connection to the broadband backbone, HP has extended its agreement with Ungermann-Bass to include the UB Buffered Repeater. Connecting the HP StarLAN Bridge to the UB Buffered Repeater connects a planning and control HP StarLAN subnet to a broadband backbone.

For production engineering, industry-standard IEEE 802.3 with ARPA and Berkeley networking services connects UNIX workstations. Production engineers can thus share files and peripherals, access mainframe resources and product design groups. Those capabilities can mean getting products to market faster -- a key benefit of CIM.

In the future, X-Windows will become increasingly important for standard graphics, multi-user and multi-application access. It will run on bit-mapped workstations, and, because of its greater flexibility, will accelerate the replacement of terminals by PCs and UNIX engineering workstations. This will increase the use of LANs, replacing terminal servers.

Production Workcell Module

A key step toward CIM is establishing effective islands of automation. Clearly, CIM is easier to achieve if these islands are designed to communicate with the rest of the plant from the beginning.

There are two major concerns in building effective workcells. The first, and generally the one that receives the most attention, is connecting the cell controller to the shop floor devices that perform the work. The second concerns the linking of workcells together in groups or areas.

HP computers commonly used as cell controllers are the HP 1000, 9000/200 and 300, and the Vectra PC. Which machine is best is a function of cell complexity, real-time needs, programming expertise and interface flexibility.

By far the most common interface to shop floor devices is EIA RS-232. Each HP computer used in cell control provides basic link level interfaces to this important standard. While not as common as RS-232, IEEE 488 (HP-IB) is more important in certain kinds of cells, such as product test and data acquisition workcells.

Once a group of workcells has been set up, users often connect the cells and establish an "area management" function. Area management is implemented to collect data from cells, store and control cell software releases, manage program maintenance and development, and to provide shared resources (discs and printers). Workcell clusters can be created in several ways: by a local subnet that links the cells with a locally owned and operated area manager, or by connecting the cells to a plant-wide backbone to access area management resources in another department or plant data center. HP AdvanceNet offers both options.

While the HP 1000 is an important cell controller today, the future of HP cell controllers is in UNIX systems. When coupled with X-Windows and low-cost PCs, UNIX will offer a powerful range of cell control solutions.

As MAP becomes increasingly well established, a low-cost link for subnets, called carrierband, will link clusters of cell controllers. Cell controllers, the equipment they control and area manager systems will be connected via this low-cost MAP subnet and gain backbone access via a carrierband-to-broadband bridge.

Computer Center Module

Getting more from a data center is easy with IEEE 802.3 ThinLAN subnets for HP and DEC equipment and SNA products for connecting to IBM and compatible mainframes. The HP ThinLAN Hub connects HP 3000s via ThinLAN coax. In addition, HP Network Services provides the capabilities to improve information access, reap the rewards of resource sharing, improve utilization of processors and communicate with other HP systems around the plant.

The future will bring more OSI/ISO networking in the form of direct MAP connections to the backbone or a subnet using TOP (Technical Office Protocol) connected via a bridge. We also expect that UNIX will begin to play a role in data center computing for off-line area management and, evenutally, more traditional applications.

Plant Area Management Module

This is the heart of HP's CIM solution. The major benefit of CIM is information access and integration, and HP AdvanceNet provides it plant-wide and among multiple vendors.

For industry-standard multivendor networking, MAP is available on some HP computer systems now, and will be on all of HP's factory systems in the future. For multivendor applications in production engineering, and for links to product design, HP offers industry-standard ARPA and Berkeley networking services for all our UNIX workstations and systems. While the ARPA/BSD services will remain important for some time, TOP will become increasingly important, integrating production and design easily into the MAP manufacturing.

Company-Wide Access Module

Here's the module that keeps a manufacturing plant in touch with the outside world. When it's important to communicate with headquarters, suppliers or customers, the alternatives within this module offer different options for doing so. In addition, HP AdvanceNet provides worldwide electronic mail, even if a customer has an SNA company-wide backbone network.

The future again includes OSI/ISO, specifically the X.400 Message Handling Services (MHS). MHS will help provide a standard foundation for Electronic Data Interchange (EDI) for connecting the factory directly to suppliers and customers.

CONCLUSION

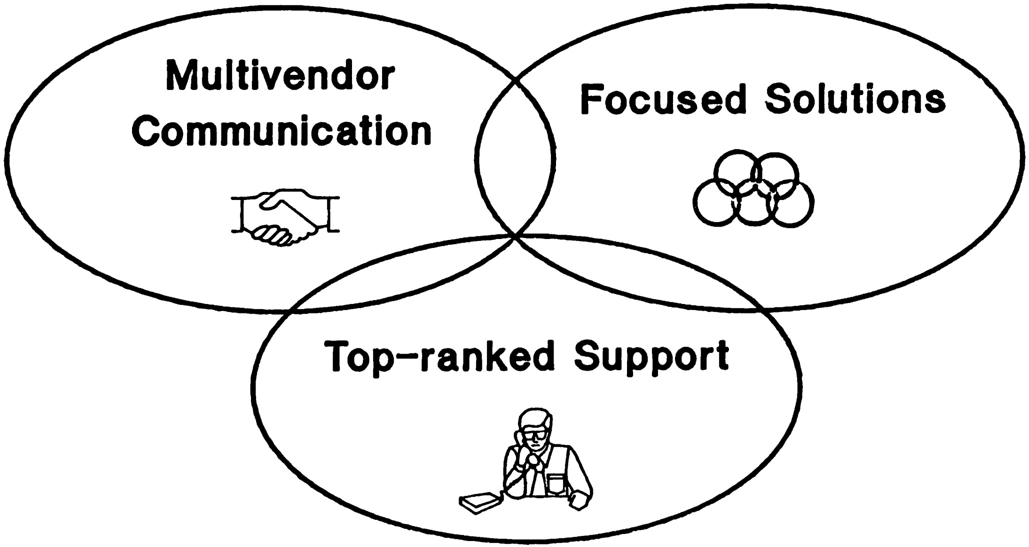
HP's CIM networking solution is divided into modules, each of which addresses a specific aspect of the manufacturing process. But it is important to note that the overall CIM solution is created by the integration of these modules. This modular structure enables users to implement their CIM solutions in manageable, affordable stages, at the rate that is ideal for them.

In the future, UNIX and DOS, linked with MAP and TOP, will become the dominant operating systems in HP's CIM solution.

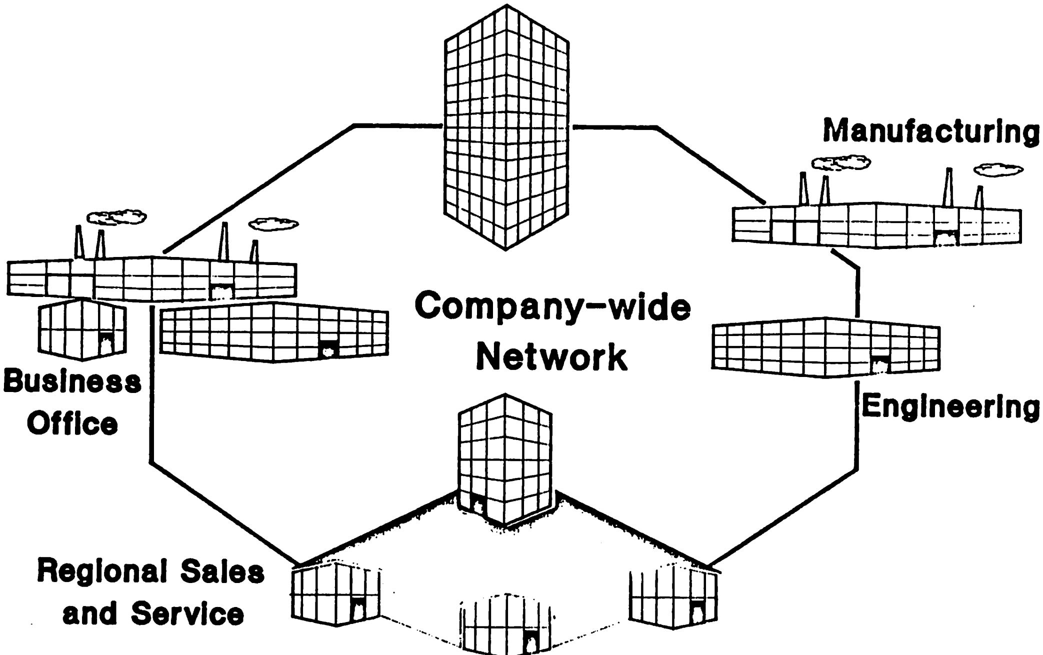
CIM is an evolving technology that has the potential to deliver dramatic improvements in manufacturing productivity and efficiency. HP has the networking experience and expertise users need to put this technology to work for them.

HP AdvanceNet

A *Strategy* for Integrated Networked Solutions



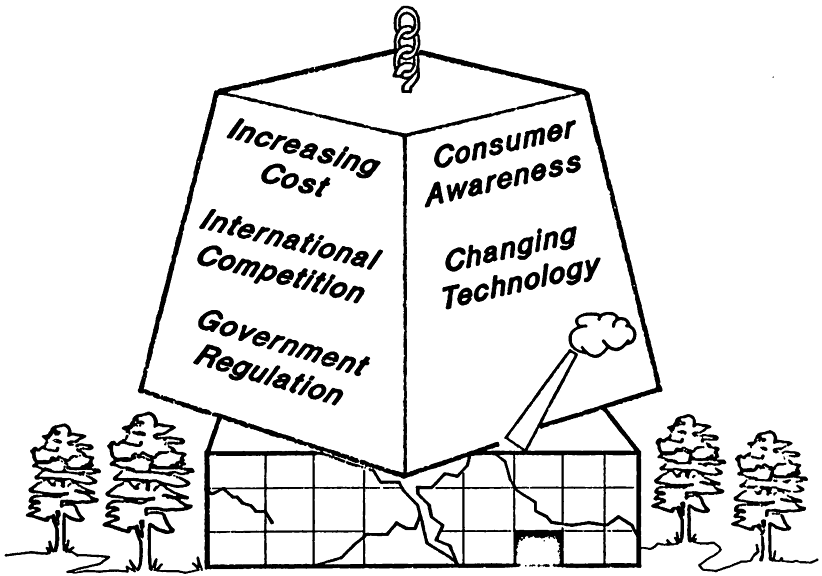
HP AdvanceNet: 5 Networking Solutions



2031-12

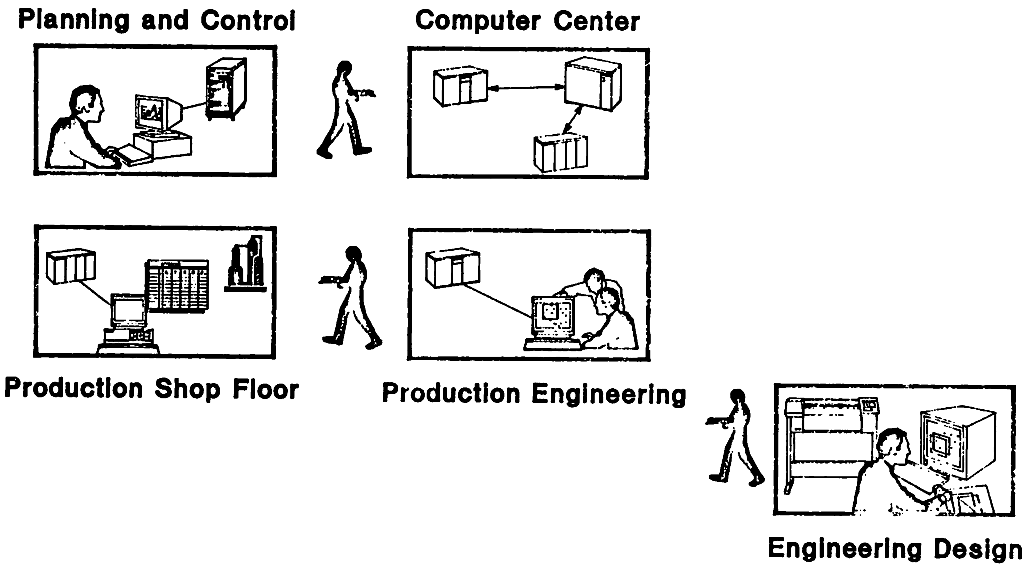
HEWLETT
PACKARD

Increasing Pressure On Manufacturing Profitability



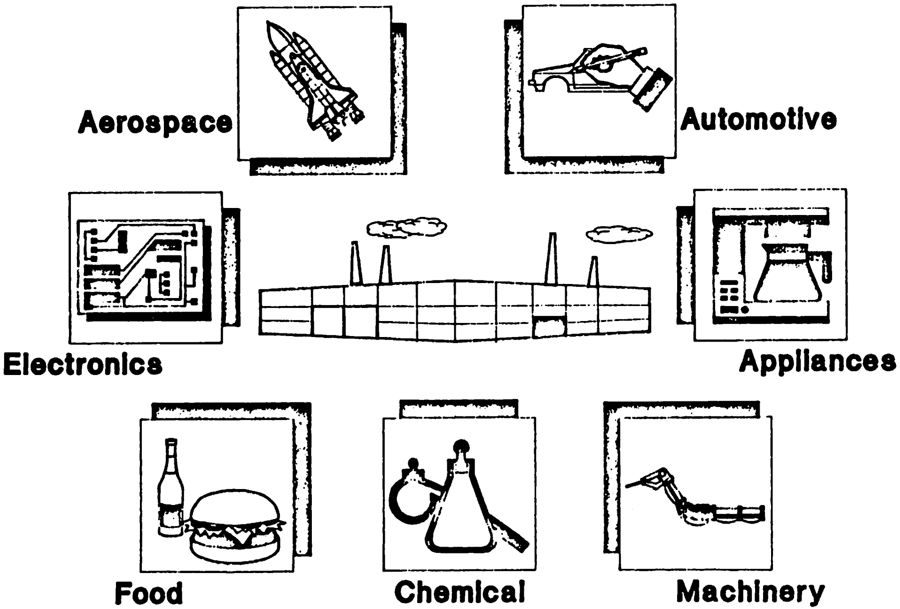
2031-13

The Manufacturing Environment



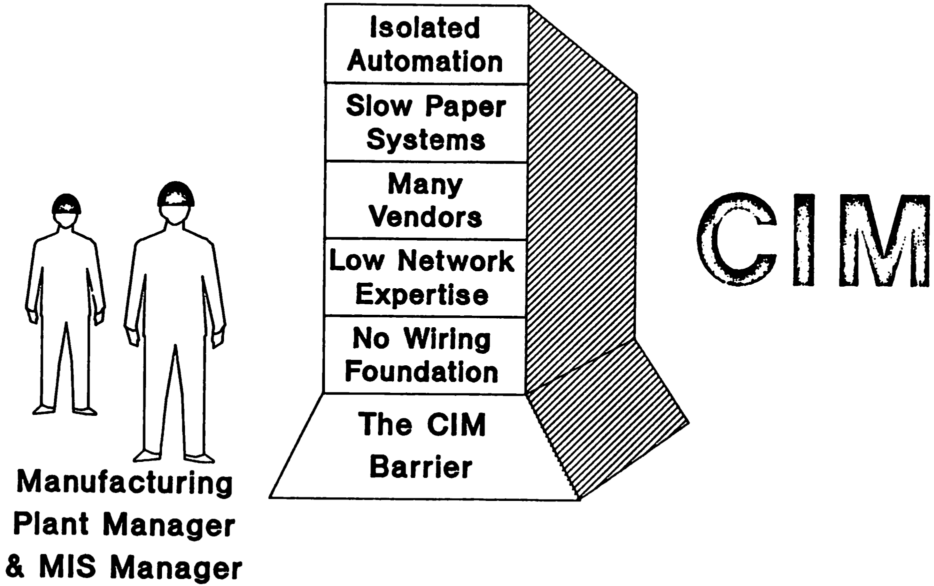
2031-14

HP AdvanceNet in Manufacturing



HEWLETT PACKARD

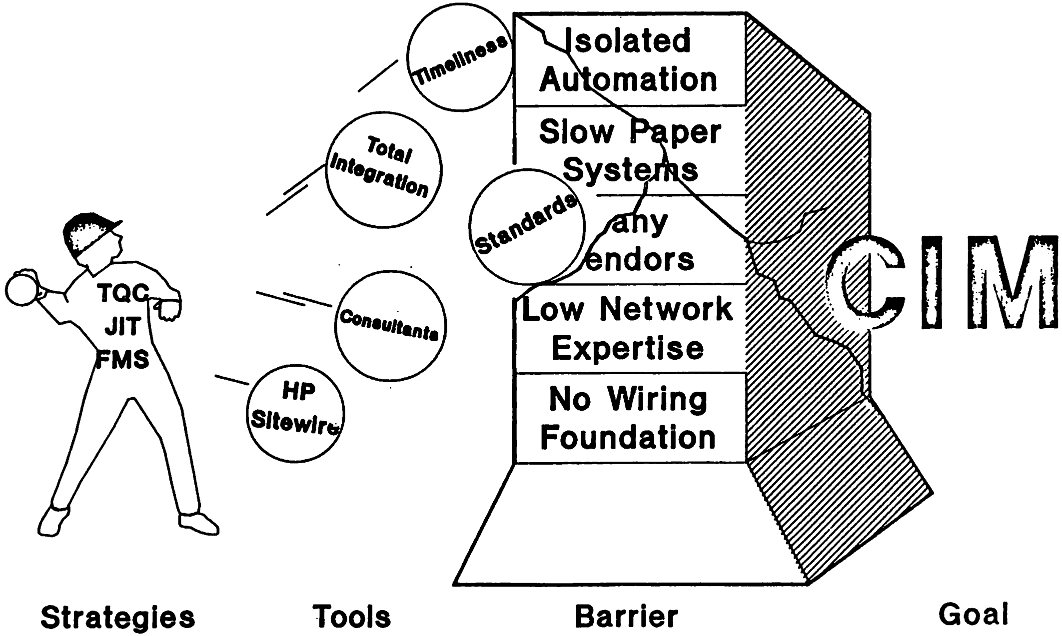
2031-15



HEWLETT PACKARD

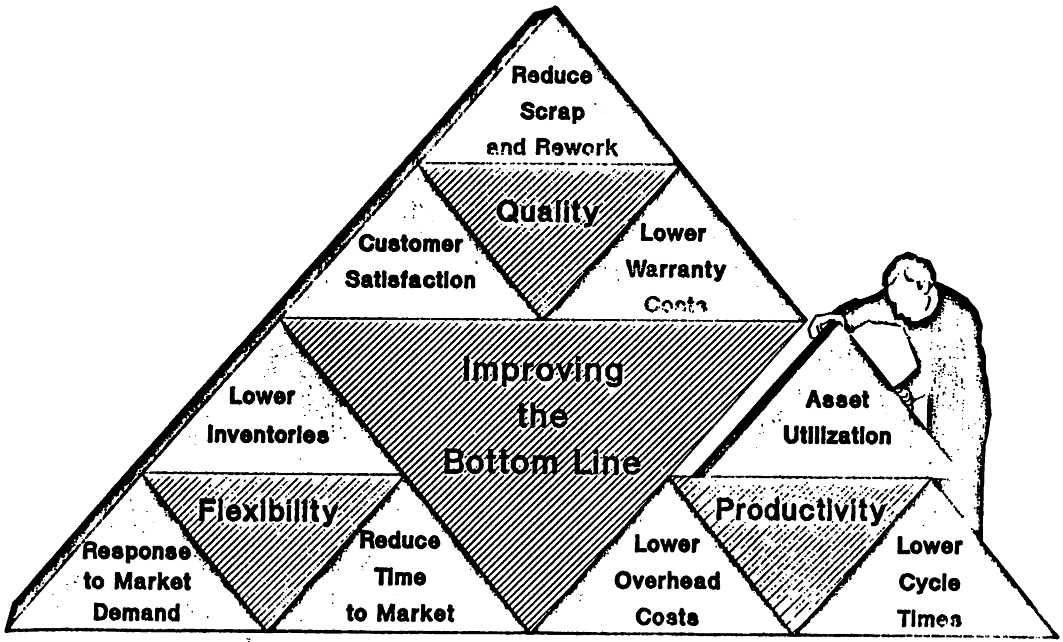
2031-16

Breaking the CIM Barrier



2031-17

CIM: Putting It All Together



CIM Networking Strategy

Consultative Expertise:
Implementation and Operation

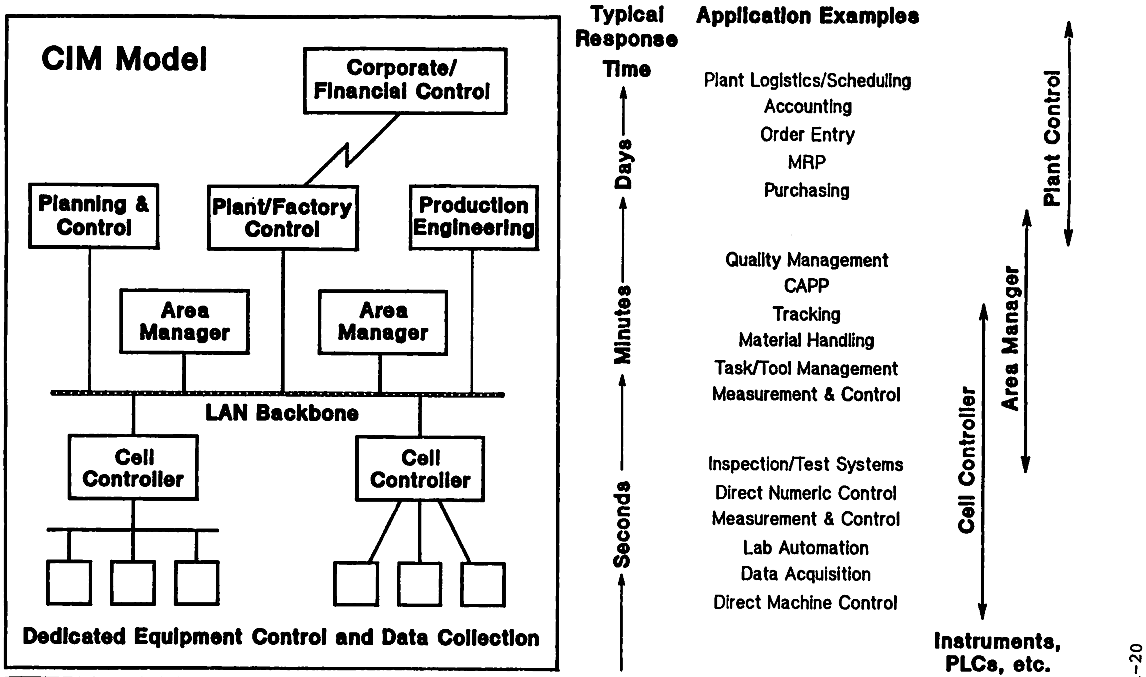
Flexible Backbone and Subnet Architecture:
A Phased Approach to Integration

Multivendor Communication
through Standards:

- MAP/OSI
- SNA
- ARPA/Ethernet

2031-19

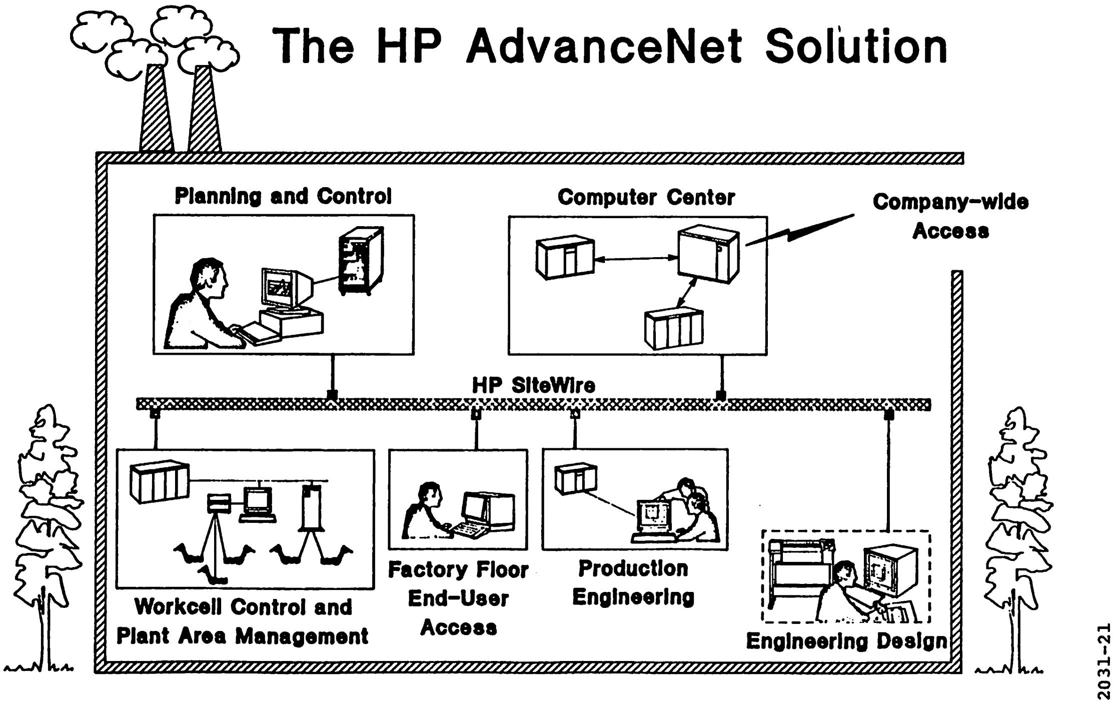
An Architecture for Success



HEWLETT PACKARD

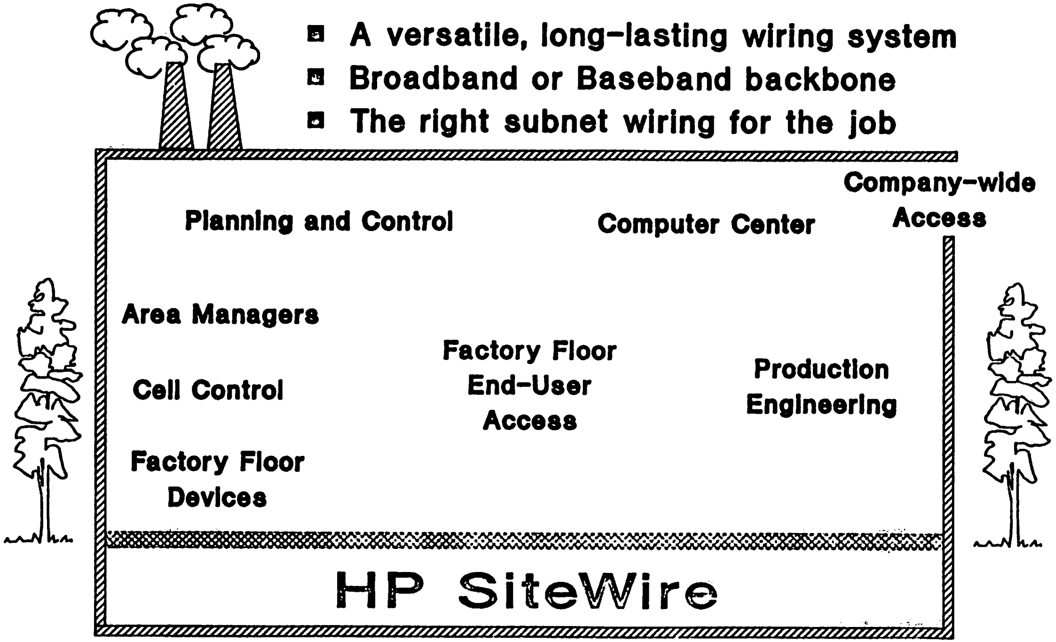
2031-20

The HP AdvanceNet Solution



HP SiteWire: The CIM Network Foundation

- A versatile, long-lasting wiring system
- Broadband or Baseband backbone
- The right subnet wiring for the job

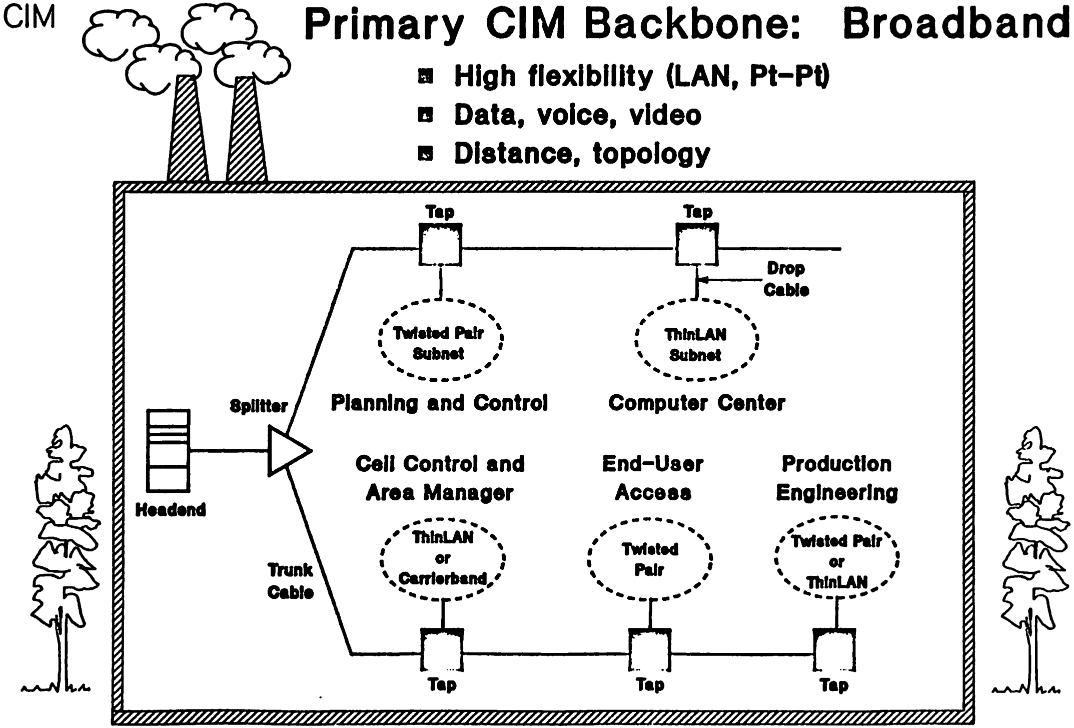


2031-22

CIM

Primary CIM Backbone: Broadband

- High flexibility (LAN, Pt–Pt)
- Data, voice, video
- Distance, topology



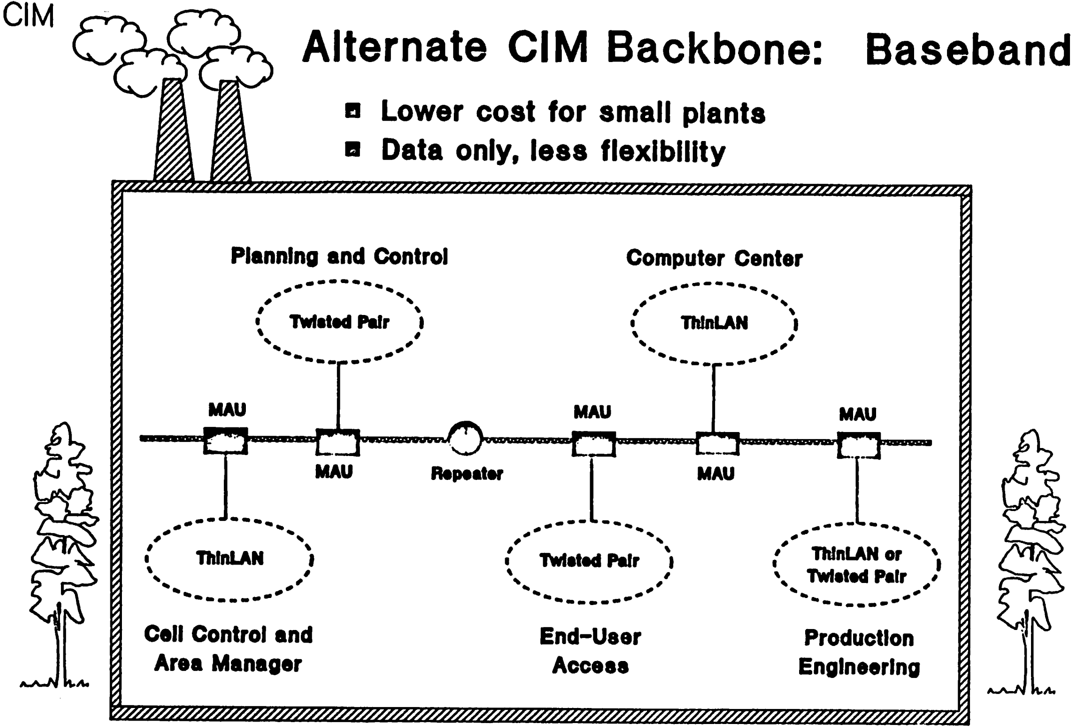
2031-23

HEWLETT PACKARD

CIM

Alternate CIM Backbone: Baseband

- Lower cost for small plants
- Data only, less flexibility

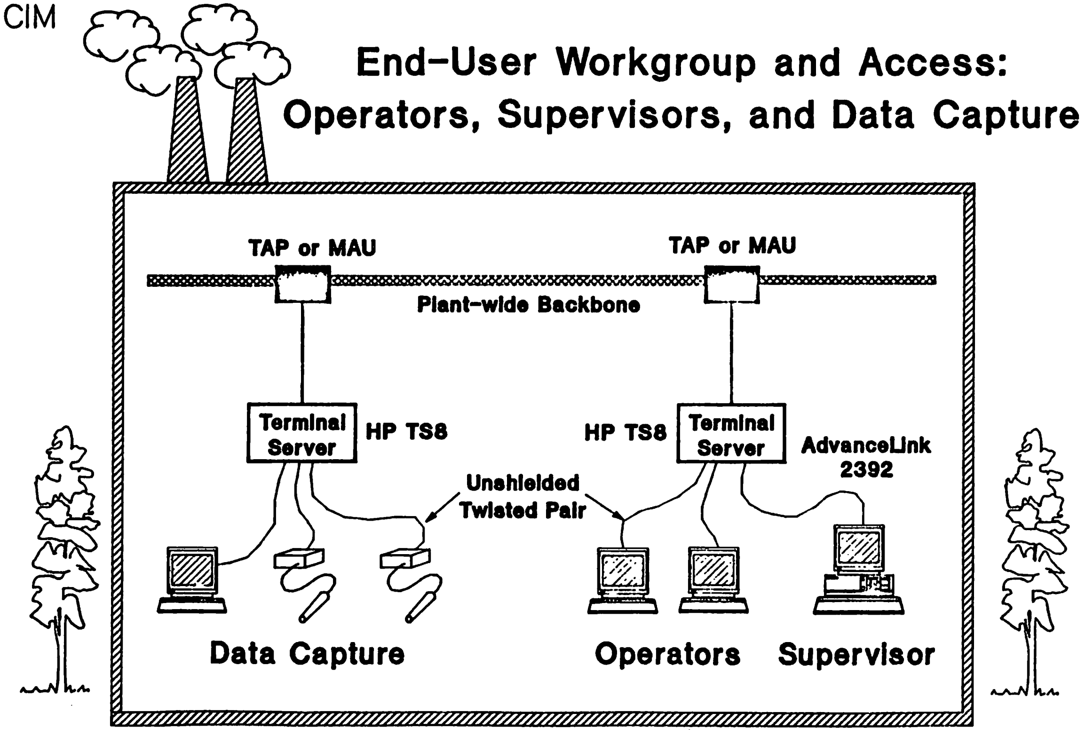


2031-24

HEWLETT PACKARD

CIM

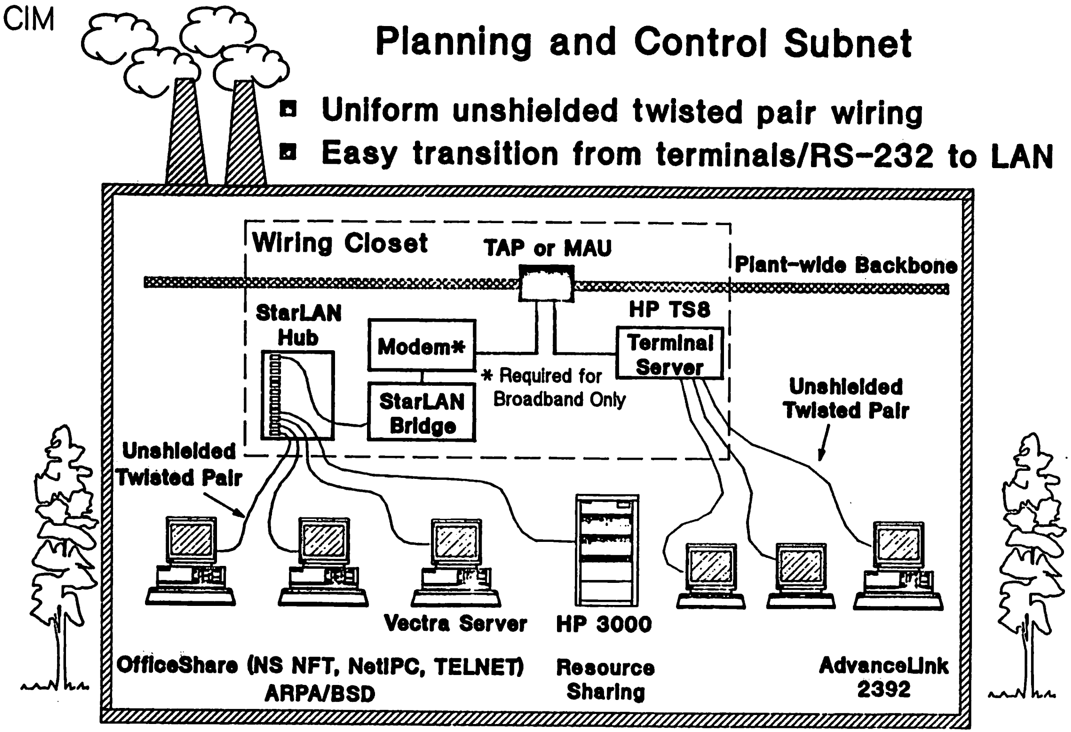
End-User Workgroup and Access: Operators, Supervisors, and Data Capture



2031-25

Planning and Control Subnet

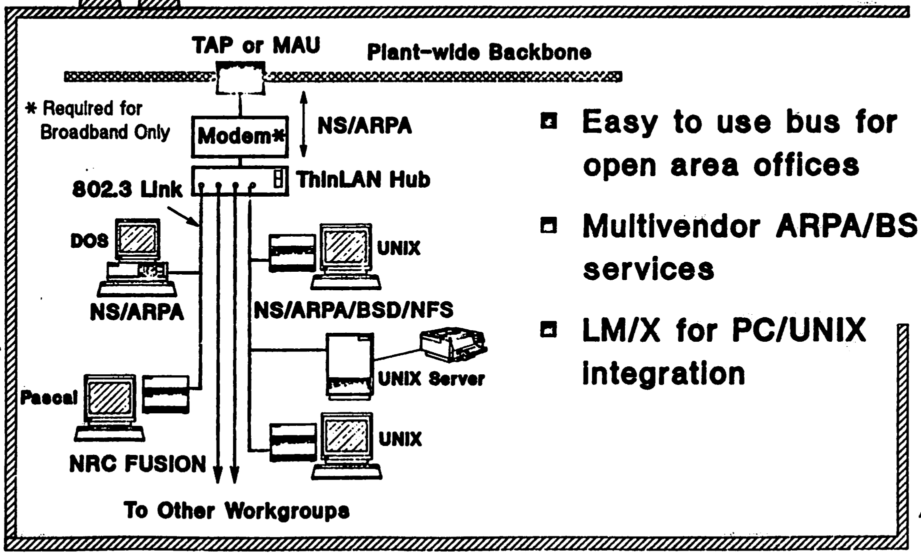
- Uniform unshielded twisted pair wiring
- Easy transition from terminals/RS-232 to LAN



2031-26

HEWLETT PACKARD

Production Engineering Workgroup ThinLAN Option



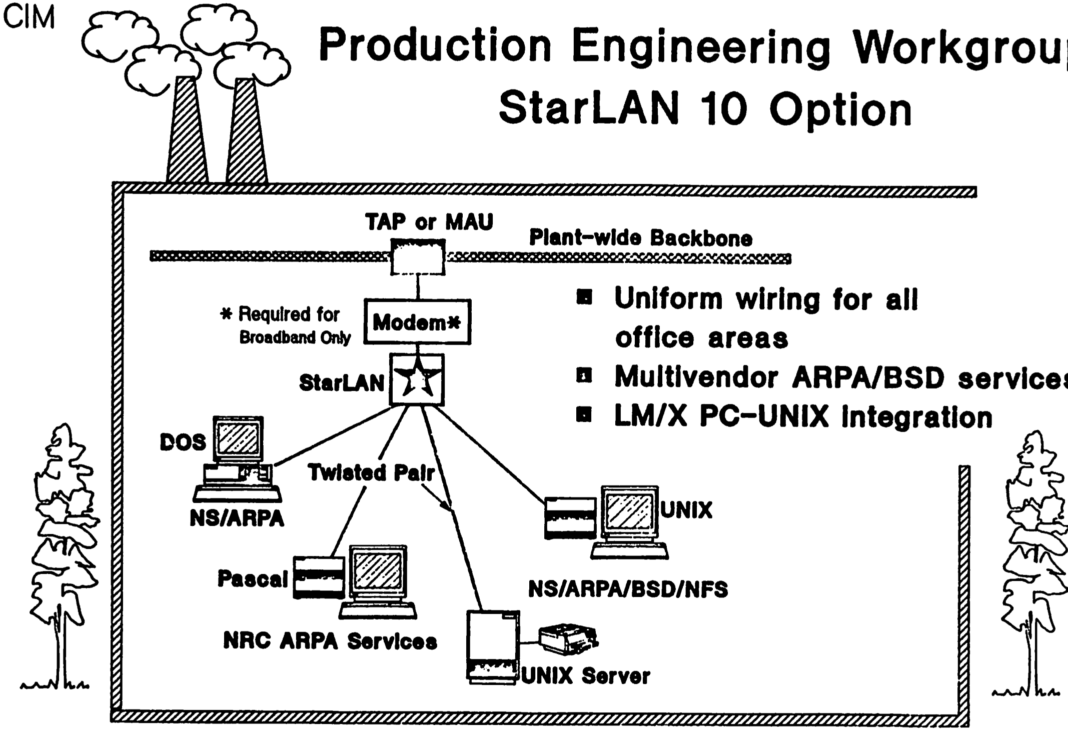
- Easy to use bus for open area offices
- Multivendor ARPA/BSD services
- LM/X for PC/UNIX integration

2031-27

CIM

Production Engineering Workgroup
StarLAN 10 Option

- Uniform wiring for all office areas
- Multivendor ARPA/BSD services
- LM/X PC-UNIX integration

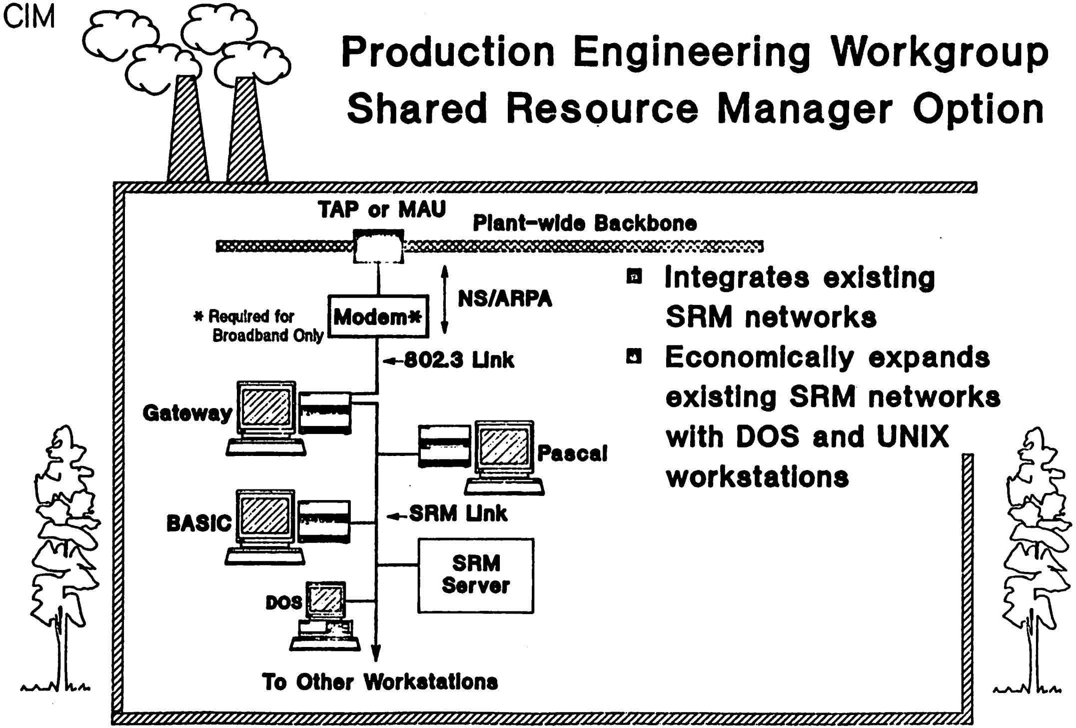


HEWLETT PACKARD

2031-28

CIM

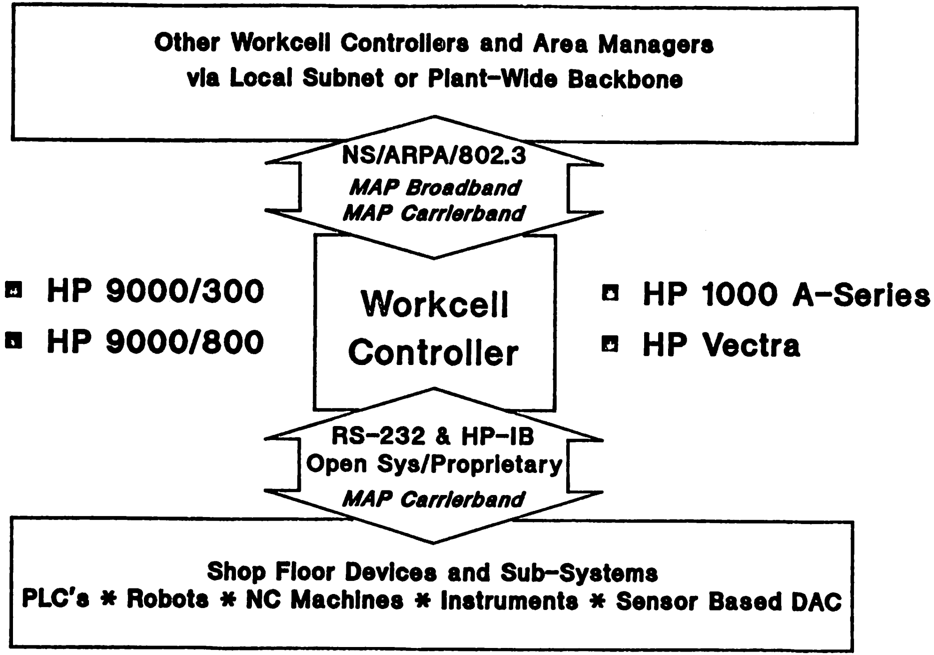
Production Engineering Workgroup
Shared Resource Manager Option



- Integrates existing SRM networks
- Economically expands existing SRM networks with DOS and UNIX workstations

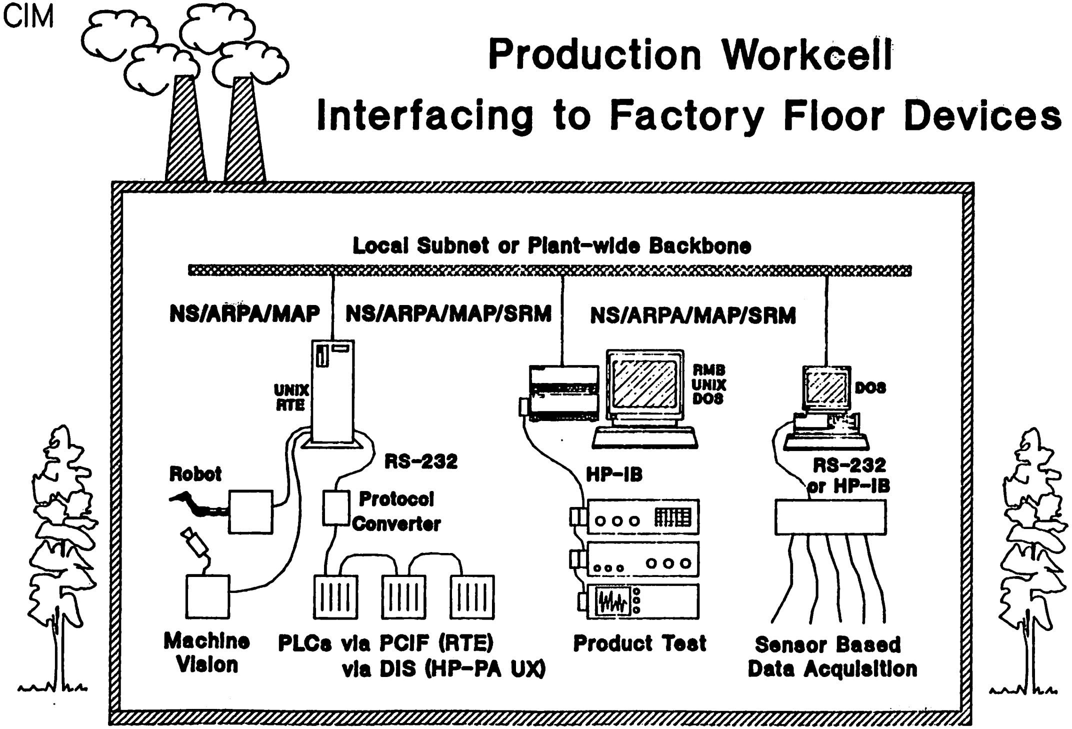
2031-29

Production Workcells



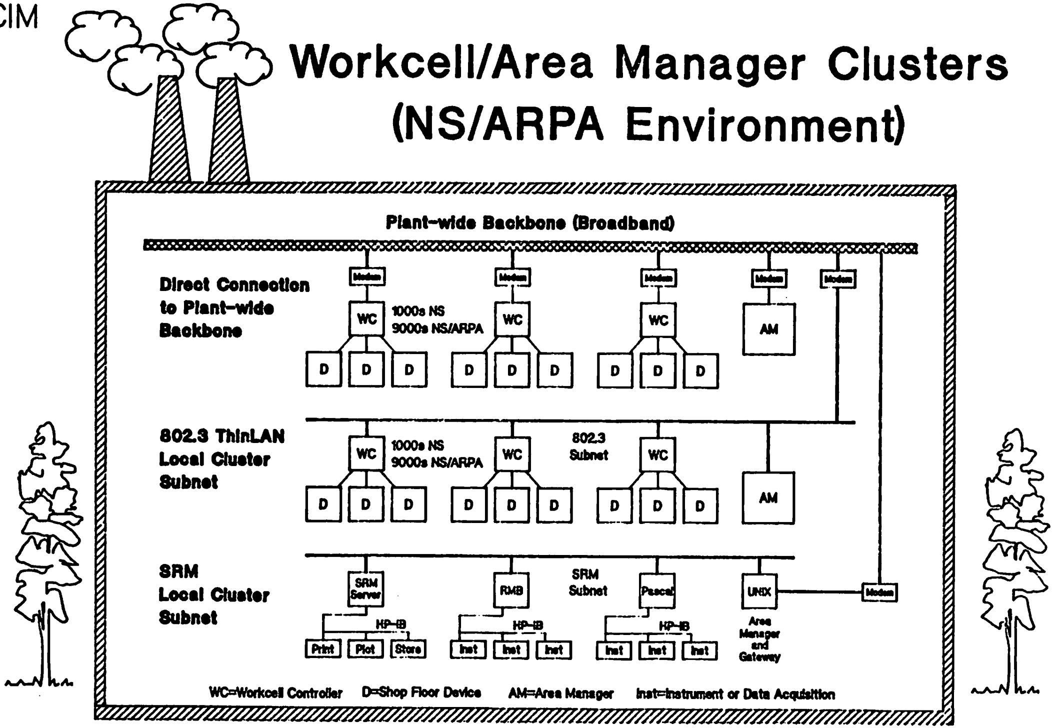
CIM

Production Workcell
Interfacing to Factory Floor Devices



2031-31

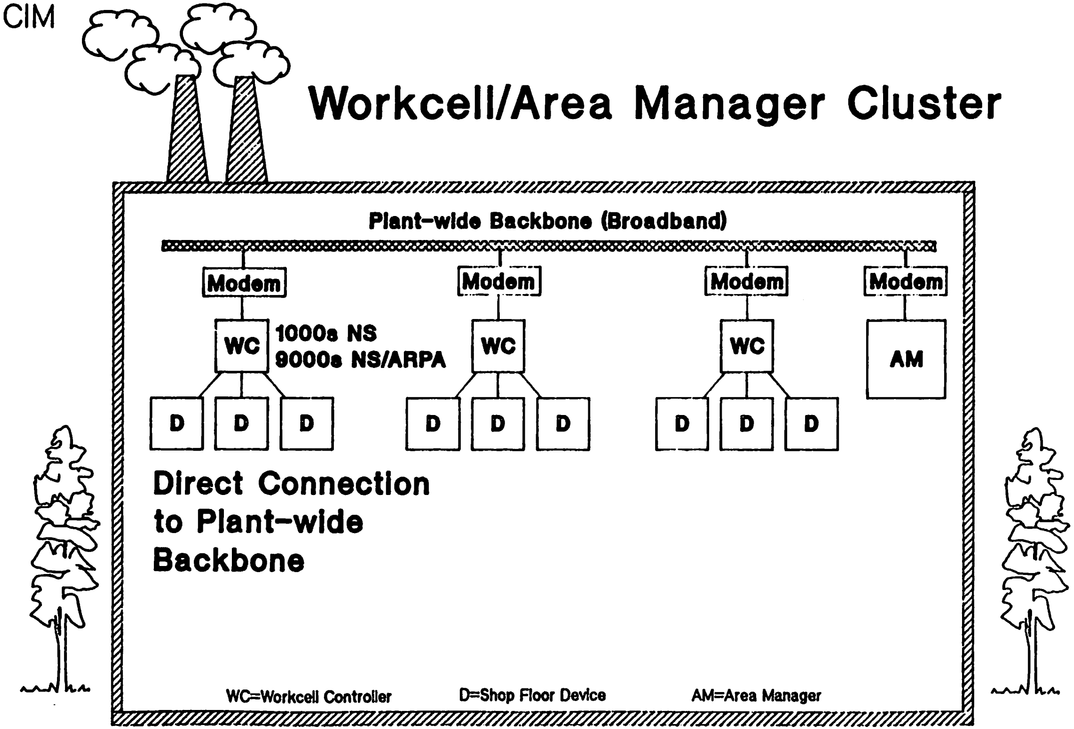
Workcell/Area Manager Clusters (NS/ARPA Environment)



2031-32

HEWLETT PACKARD

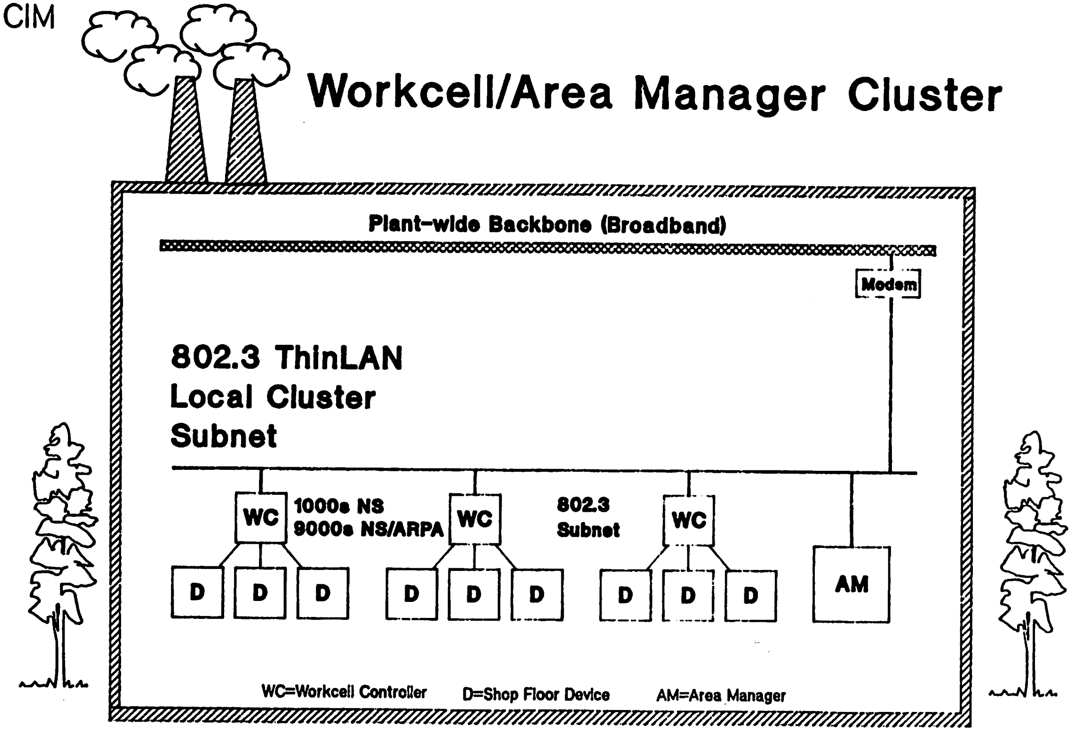
Workcell/Area Manager Cluster



2031-33

CIM

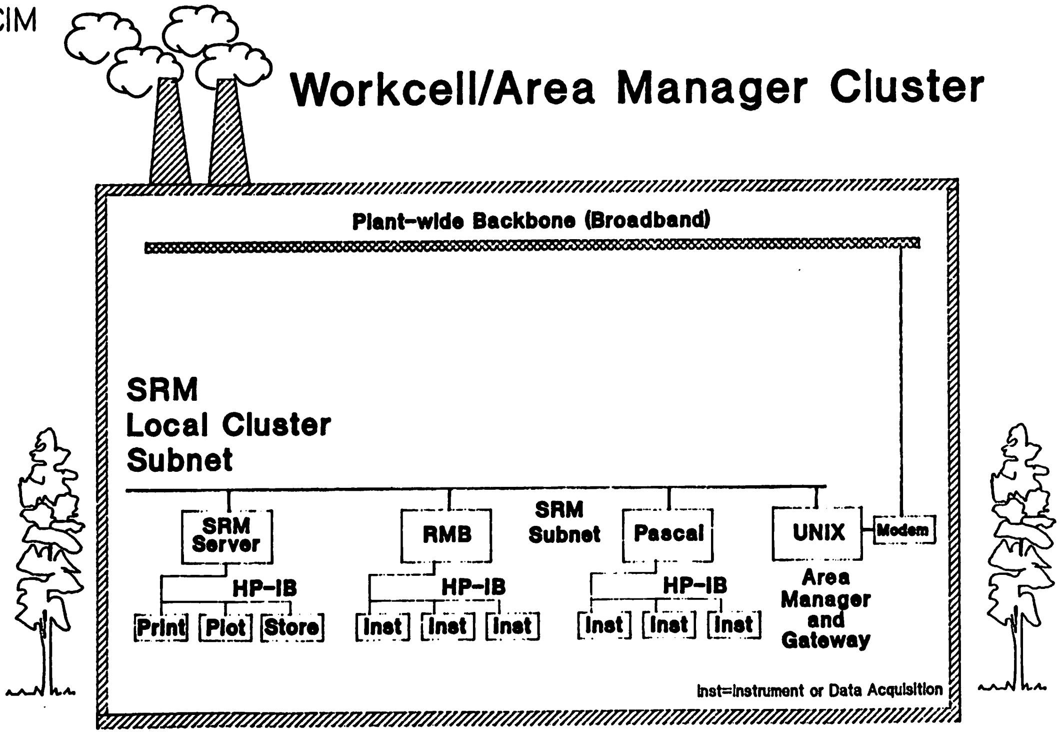
Workcell/Area Manager Cluster



2031-34

CIM

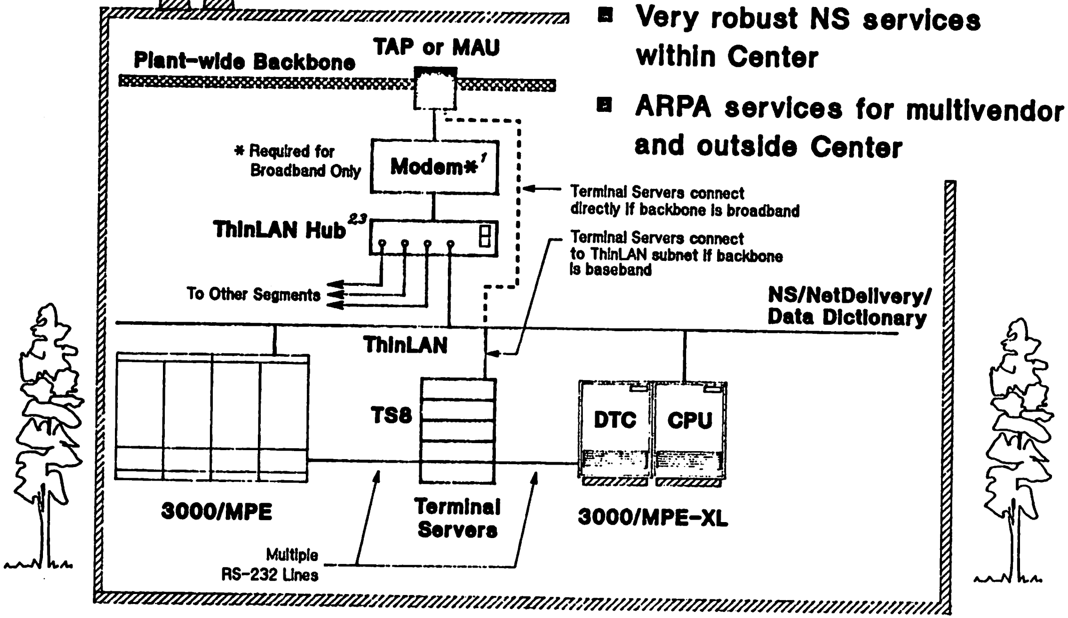
Workcell/Area Manager Cluster



2031-35

Computer Center: MPE

- Very robust NS services within Center
- ARPA services for multivendor and outside Center

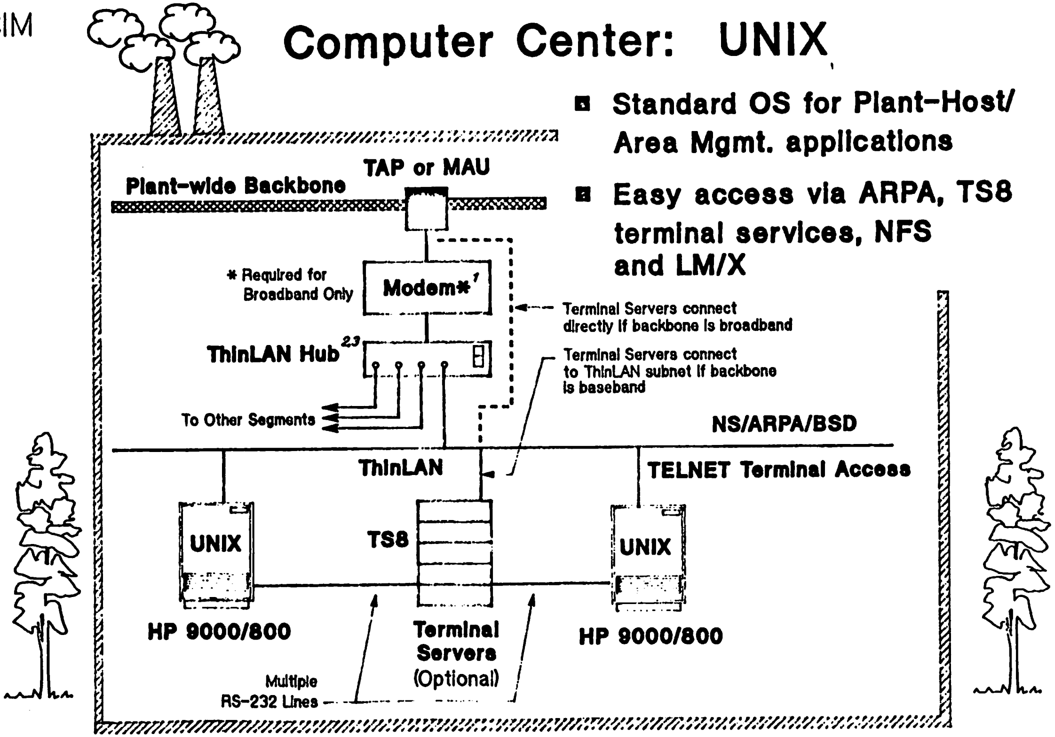


2031-36

CIM

Computer Center: UNIX

- Standard OS for Plant-Host/ Area Mgmt. applications
- Easy access via ARPA, TS8 terminal services, NFS and LM/X



2031-37

HEWLETT PACKARD

Integration:
Intersystems Networking At-a-Glance

	HP1000 A-Series	HP 3000 MPE	HP 3000 MPE-XL	HP 9000 300-UX	HP 9000 PA-UX	Vectra	Other Vendors
HP NS *	✓	✓	✓	✓	✓	✓	✓ DEC
ARPA	Future	Future	CF	✓	✓	✓	✓ SUN, DEC
Berkeley				✓	✓	✓	✓ SUN, UX
MAP 2.1	✓S	✓S		CF	✓S	CF	✓ COS
MAP 3.0			Future		Future	Future	✓ COS
SNA	✓S	✓	✓G/N	✓G/N	✓G/N	✓	✓ IBM
Bisync	✓	✓					✓ IBM

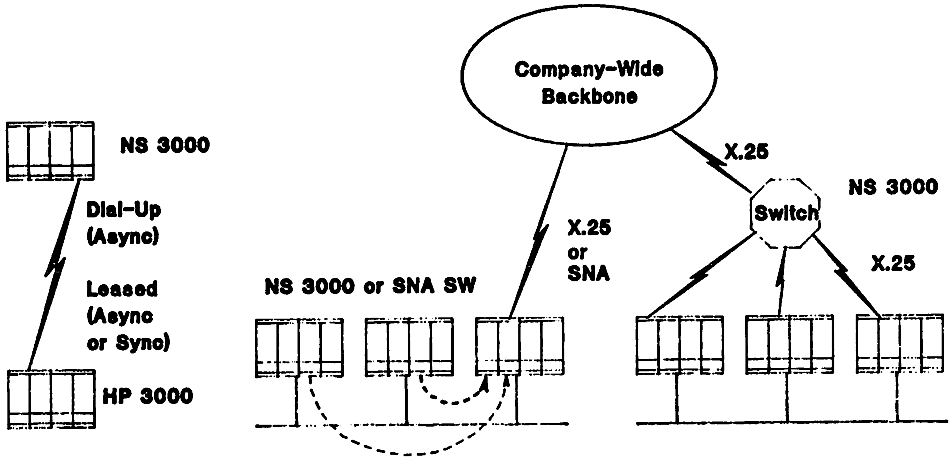
G/N = Gateway/Native S = Special Quote CF = Consult Factory

COS = Corporation for Open Systems Conformance and Interoperability Support.

* Services supported varies by processor. Refer to Intersystem Matrix.

CIM

Company-Wide Access



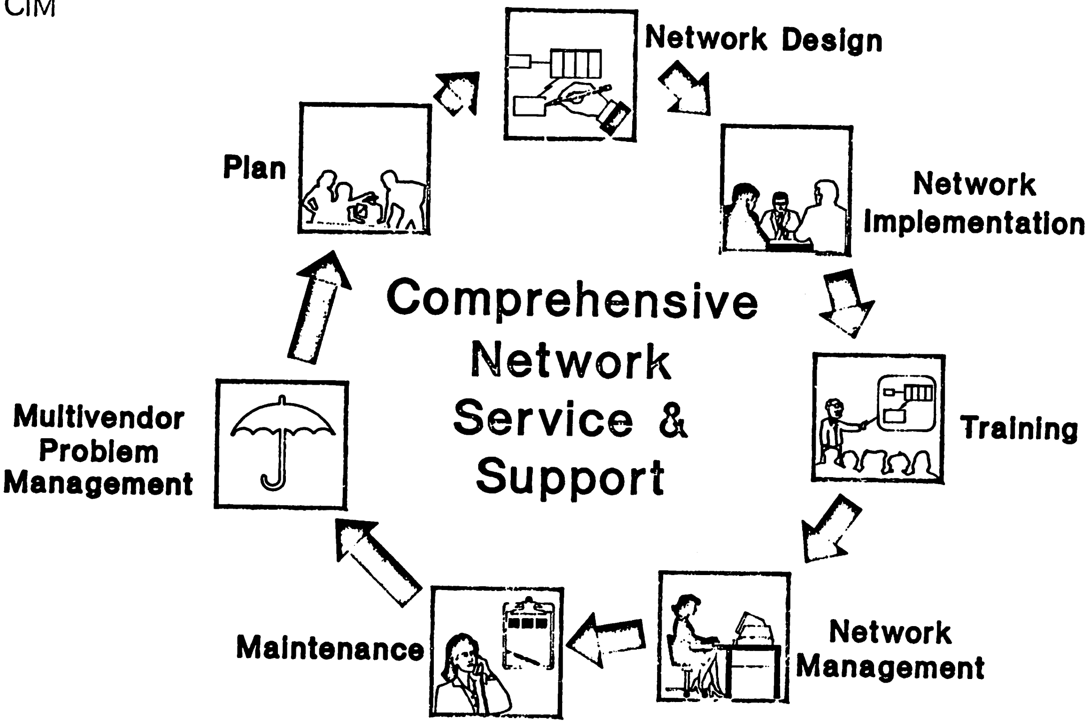
Point-to-Point
Small Nets

X.25 or SNA Gateway
Minimize Access Cost

X.25 Direct
High Traffic, High Performance

2031-39

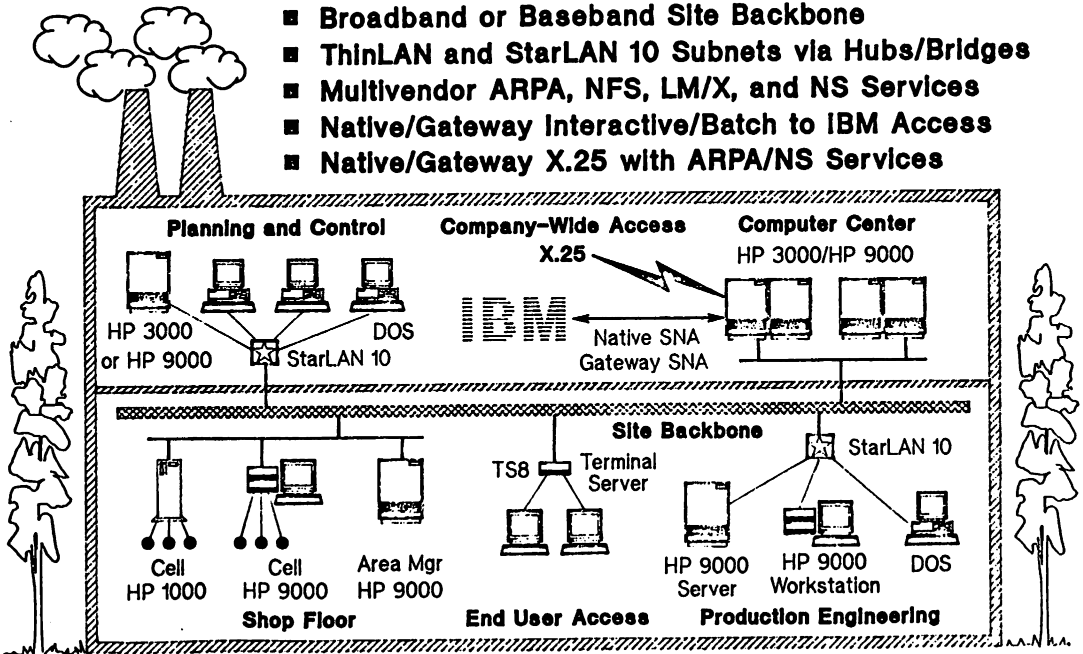
HEWLETT PACKARD


Network Design
Network Implementation
Training
Network Management
Maintenance
Multivendor Problem Management
Plan
Comprehensive Network Service & Support

2031-40

Putting It All Together – ARPA/NS/802.3 Environment

- Broadband or Baseband Site Backbone
- ThinLAN and StarLAN 10 Subnets via Hubs/Bridges
- Multivendor ARPA, NFS, LM/X, and NS Services
- Native/Gateway Interactive/Batch to IBM Access
- Native/Gateway X.25 with ARPA/NS Services



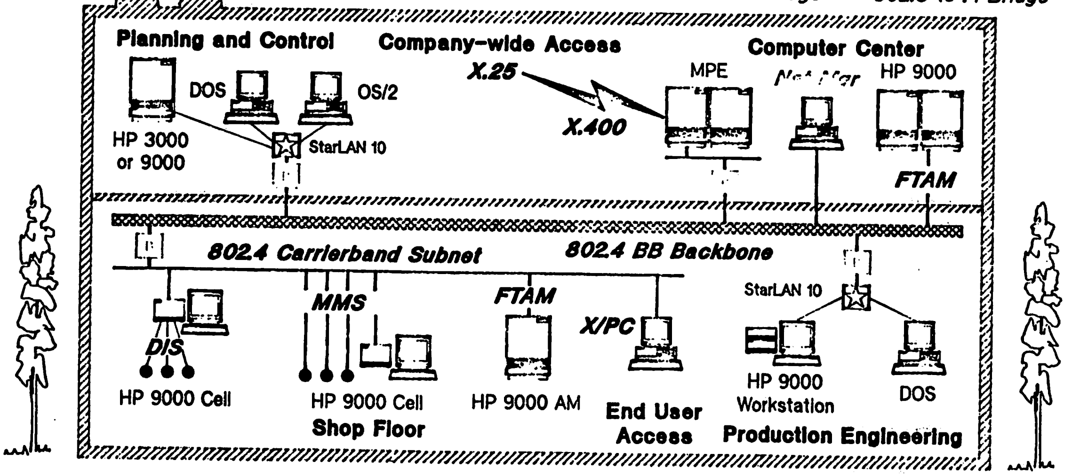
Putting It All Together with MAP/OSI

Beginning In Late '88

- HP MAP 3.0 MMS Software
- HP MAP 3.0 FTAM Software
- HP OSI Express 802.4 Interface
- HP Device Interface System (Non-MAP Devices)

Plus Statements of Intent for 1990 Support of:

- *PCI*
- *Network Management*
- *802.4 CC/BB Bridge*
- *802.3 to .4 Bridge*



HEWLETT PACKARD

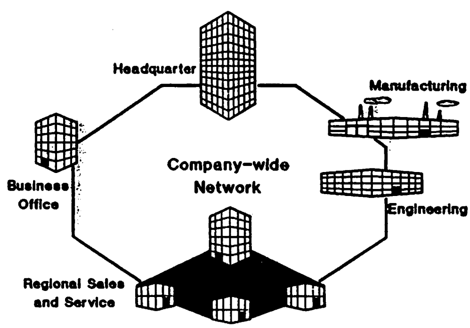
HYBRID NETWORKS
PUBLIC VERSUS PRIVATE PACKET NETWORKS
THE BEST OF BOTH WORLDS

Patrick LELORIEUX
HEWLETT-PACKARD
GRENOBLE NETWORK DIVISION
19420 Homestead Road
Cupertino, CA 95014

COMPANY ENVIRONEMENT

Maintaining the competitive edge in todays'changing market place is becoming more and more dependent on efficient management of information flow. Getting the right information to the right people at the right time means cutting operation costs, boosting productivity and increasing overall customer satisfaction. Keeping abreast with datacommunications technology ensures that your organization maintains that competitive edge.

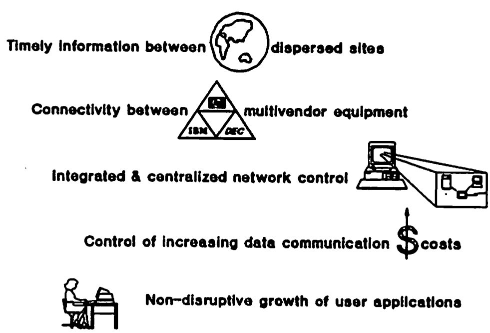
A typical company wide network environment reflects geographical dispersion, equipment from multivendors, rising datacommunication costs and integrated applications.



WIDE-AREA NETWORKING NEEDS

Very often company-wide network, due to a decentralized approach or an approach tailored to sysyem applications, results in the establishment and operation of sevearal independant data networks. Each network is optimized for a particular application and is usally incompatible with the other networks. Within such an environment the company expects multivendor connectivity, integrated network control, security, reliability and cost control.
The company-wide network sould have the ability to provide connection of any data processing equipment (IBM, DEC, HP ...), should be able to share the network facilities for intra vendor comunications (HP to HP, IBM to IBM, ...) as well as for transmitting information between each other.
With the wide area network becoming such a crutial element of the company's business, the company wants to retain integrated and centralized control of the network.
The network should offer sufficient modularity and flexibility to enable the company to optimize the cost of datacommunications. Cost is generaly not the initial motivating factor behind a wide area network program, but it rapidly becomes important as the investigation starts. Datacommunications costs are spead out over the entire company and are relatively insignificant compared to voice communications costs, however looking closer, more alarming is the growth rate of the datacommunications costs which is estimated to thrirty to forty percent each year.
The wide area network is the backbone of the company allowing various entities to communicate. Without a very reliable network the efficency of the company will be strongly affected.

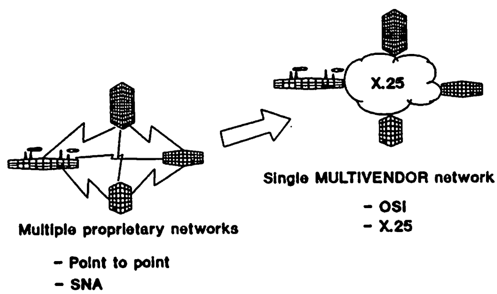
Wide-Area Networking Needs



COMPANY WIDE NETWORK ALTERNATIVES

In our current market situation, emphasis is placed on dedicated networks that incorporate multiple proprietary architectures with point-to-point connections and access to public data networks (PDN). The trend, however, is moving towards a multivendor solution on a single network using a backbone based on the X.25 standard. X.25 offers a reliable protocol based on industry standards, full routing capability, and optimal use of transmission link capacity.
As a Yankee Group report said, "Packet switches are and will increasingly become the network node of choice for large organisations with the need to connect greographically disrtibuted computers from a variety of vendors".

Company-Wide Network Alternatives



X.25 PUBLIC VERSUS X.25 PRIVATE

It is however important to keep in mind the differences between a public packet switching network and a private one, and the benefits of each. Public packet switching networks are supplied to the public by the local PTTs in each European country,and by VAN's in the US. The PTTs or VANs supply telecommunication facilities and provide packet switching for computer networks.

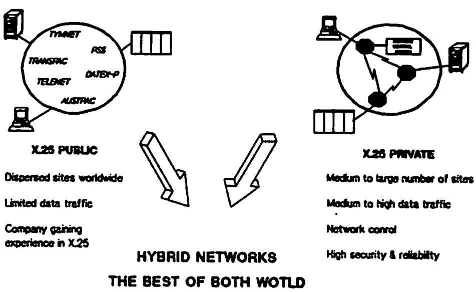
Private packet switching networks are designed and implemented by private organizations, and are tailored to the organization's specific needs. The transmission facilities needed to interconnect the switching nodes, users, and computer systems are normally leased from the PTT's.

But for most users, the issue is not public versus private packet switched network. The question is rather how to combine the features of both into a network that is tailor to specific requirements and conditions. Such hybrid networks are often the solution.

Both public and private networks offer specific strengths that a hybrid network can selectively incoporate. A public packet network reaches low-density sites which are widely dispersed more cheaply than a private network does. In addition, the public network can be used as a back up for private packet network peak loads or failure. Public networks also relieve operators of the responsabilities for network administration, access control and user authentification.

Private packet networks have their own advantages. They offer users centralized management and control that is tailored to the organization needs. Private packet network being based on X.25, insures multivendor connectivity and easy coexistence with SNA environments. An X.25 private network also provides high connectivity for remote sites, which makes it easier to interconnect an entire organization.

Backbone Network
X.25 Public Versus Private



Let's look in more detail at how public and private packet networks compare.

1) Connectivity : As the Gartner Group report explains, private packet switching offers better connectivity by providing easier terminal-terminal and terminal-host connectivity, wider range of protocol conversion, and ensures effivcient use of costly long-haul facilities.

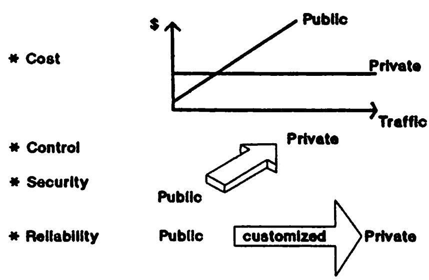
2) Control and Security : A private network means better control and security for the network. Superior network management and security are often what motivates users to move to private packet switching. Public networks typically address the issue of security by setting up procedures for verification and access control. These procedures are however administrated by the local PTT's, not the users. Control of planning, design, availability, cost, security and capabilities are inherent in private packet networking.

Public packet switching offers several operational pluses. Users do not have responsibility for operation or maintenance of the network, nor are they responsible for data integrity. The public network offers the ability to link up with multiple locations, and usually does not require a lot of initial expenditure.
Alternatively these pluses also have drawbacks, specifically lack of controlis an issue. A private packet switching network offers increased security, reliability and full network management control as well as customized design for cost effectiveness, availability and performance.

3) Reliability : Businesses whose operations span the globe need continuous operation of their networks. Public networks reliability varies from country to country, and provides standard reliability for all location. The configuration of a Private network, with an appropriate network design, tailors the reliability to the company needs, ensuring high availabilty in the critical parts of the network.

4) Cost-effectiveness : The rising costs of datacom lines make the advanced X.25 switching technology very cost effective compared to traditional point-to-point links. Although public networks provide X.25 switching technology, when a certain volume of traffic is generated it becomes more cost effective to lease dedicated lines and install your own private network. PTT's tariff charges inside a public private switching network are based on connection time and volume and not on distance covered. Thus typically companies with a large number of sites spread over great distances especially internationaly with medium traffic are prime candidates for a private packet network.
With medium-to-high volume traffic, the advantages of private packet switching play a large role. These systems blend the multiplexing of public packet switching with the time and volume independence of private circuits. Control of all aspects of the network sets private packet switching apart as an alternative.

Public Versus Private



These guidelines do not exhaust the questions to be considered in implementing a packet switching network, but they do provide a sound basis for planning.

BEST OF BOTH WORLD

It is a tradeoff -- with a public network, users are buying services ; in a private packet switching network, they buy the entire system.

Hybrid networks, which combine private and public networks, offer the ultimate in flexibility and are increasingly the choice of businesses that rely heavily on their networks. A hybrid enables an organization to set up its own private network, but use public services for connectivity to very remote locations with low traffic volume.

These advantages are expected to become increasingly important to all types of organizations in the next few years. Industry analyst groups that have studied the market see a bright future for hybrids of private and public networks that are tailored to the specific needs of their users. As hybrid systems become more common, many large multinational organizations will learn what many Fortune 100 companies have known for years.

Since both private and public packet switching have their pluses, it is logical to seek to combine them in a way that maximizes these trends. That is what is driving the growth in hybrid networks. Connecting private X.25 networks to the public X.25 networks is what companies including Citibank, Bank of America and Shell have done in the last several years. The advantages of hybrids -- superior network managment and control, better load sharing and network availability from private, reduced costs from a mix of private and public -- are attractive to many companies.

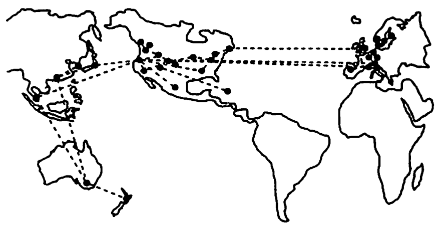
Hewlett-Packard EXAMPLE

Among those companies is Hewlett-packard, the international manufacturer of computer systems, test and measuments and scientific instruments. The company employs 82,000 people worldwide, and in the early '80s its internal networking costs were rising steeply.

HP decided to implement a hybrid network based entirely on X.25 technology in several phases. The network connects 2,500 systems, uses 40 backbone switces and handle both batch and interactive traffic. Electronic mail, sales order administration, engineering (CAD/CAM) as well as pushasing and inventory management are the applications running over the network. The company's investment in its revamped corporate network was just under $6 Million. In the time it has been operational, cost savings have been running at almost $2 million per year, which means that the payback period is a mere 3-4 years.

HP's Hybrid network has met other company objectives in addition to cutting costs. It has contributed to improvements in the assembly of and access to financial and analytical data, reductions in production cycle times, shortened product developement cycles and enabled the company to reduce overall inventories.

Hewlett-Packard Company Network



Networked Entities	X.25 Network	Applications
• 82,000 Employees	• 25 Backbone switches	• Electronic mail
• 438 Offices	• 150 Stand-alone switches	• Sales order administration
• 70 Countries	• 2 Management centers	• Purchasing/inventory management
• 2,500 Host computers	• 120 Giga bytes / month	• Engineering (CAD/CAM)
• 73,000 Workstations	• 1.3 Million calls / month	• Remote database access
• 600 PADs		

Telecommunications vendors were the first to offer packet switching services, and many have started to add private services. Now computer vendors, with Hewlett-Packard first into the market place, are offering such services as well.

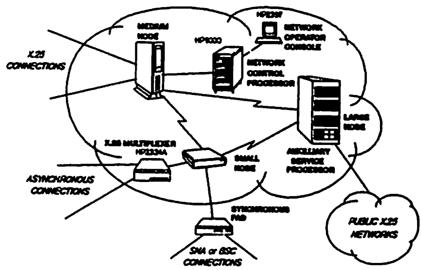
HP PRIVATE PACKET NETWORK OFFER

In early 1987 the Hewlett-Packard Company announced that its company-wide networking solution would be based on packet-switching. HP had decided that packet-switched networking tailored to customer needs, delivers the optimal networking solution.

The HP Private Packet Network (HP PPN) is designed to provide a highly reliable packet capability of connecting devices having standard CCITT X.25 packet interfaces, and non-packet mode devices through Packet Assembler Disassemblers (PADs).

With HP X.25 PPN you can build your own dedicated private X.25 network. It is composed of a wide range of switching nodes , a Network Control Processor and Auxiliary Service Processors that manage all network activities, through Network Operator Consoles. Its modularity insures a cost-effective implementation for small as well as very large networks.

HP PRIVATE PACKET NETWORK



The key benefits and unique product features are as follows :

* CONNECTIVITY

The HP PPN provides the general utility network. The following are included:

- X.25 Connections - High speed up to 64 Kbps
- X.25 Gateway for access to public or other private networks
- Asynchronous, SDLC and BSC protocols support, through PADs

* NETWORK CONTROL

The Network Control System allows effective network management from a central site or through remote operator consoles.

- Easy-to-use, forms-based operator interface
- Configuration control - Online, offline, real-time changes
- Automatic statistics collection
- User definable report generation
- Event and alarm filters for rapid isolation of failures
- Centralized capabilities which provides robust control

* RELIABILITY

The HP Private Packet Network features both hardware and software redundancy.

- Redundancy within the large nodes - sparing of the switching boards, multiple buses, and redundant power supplies
- Online Service - switchover is automatic
- Dual Control Processors can be distributed for disaster backup, independant operation and databases duplication to insure system integrity, fall back capability, ease of network expansion
- Auxiliary Service Processor in the large nodes further enhances reliability and performance
- Adaptative routing provides transparently to the user, call rerouting upon link or node failure or if more cost effective path available

* SECURITY

The HP Private Packet Network is protected against unwarranted use and access.

- Internal access restrictions done by security checking on network access points, both on source and destination addresses, with time and days filters
- Access restriction to Public networks
- Multiple user classes, which can restrict routes used, and give priority to types of traffic

* COST EFFECTIVENESS

Due to its advanced technology, HP PPN minimized communication costs .

- Wide range of switching nodes, from 8 to over 500 ports with modular design providing nodes upgrate capability
- Three types of Nodes - small node Model 60, medium node Model 70, and large node Model 80, Model 70 expandable to Model 80
- Built-in redundancy which reduces maintenance costs, and "out-of-service" losses
- Dynamic Routing ensures least cost routing and best efficient use of link resources
- Online Remote Reconfiguration reduces personnel costs, changes take minutes rather than days to perform.

* EASE OF USE

The HP Private Packet Network offers high flexibility.

- "Hot" module replacement enables boards removal while equipment continues normal switching operation
- Logical X.121 addressing allows easy network changes, address plan being independant of network topology.
- Menu-driven operator interface, network control no longer requires specific command language

HP PRIVATE PACKET NETWORK

FEATURES & BENEFITS -

- X.25, ASYNCHRONOUS, IBM CONNECTIONS → MULTIVENDOR CONNECTIVITY
- MENU-DRIVEN ADVANCED NETWORK MANAGEMENT → NETWORK CONTROL
- BUILT-IN REDUNDANCY → HIGH RELIABILITY
- WIDE RANGE OF SWITCHING NODES → COST EFFECTIVENESS
- DYNAMIC ROUTING → RELIABILITY / COST EFFECTIVENESS
- MODULAR DESIGN, EASE OF CONFIGURATION → FLEXIBILITY

⇨ DURABLE SOLUTION

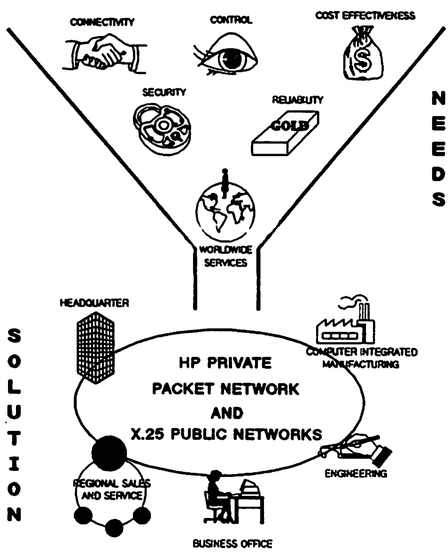
SUMMARY

"How networking helped one industry cut paper mountains to bits" (Data Communications).
"Switch to private net satisfies users's need for control" (Computer World).
" Packet switching is firm's link to greater productivity" (Data Communications).

Reading and hearing about such success stories from our industry peers, suppliers or even competitors, are powerful incentives to get start on a similar project. More and more companies are discovering the benefits of wide area networking. A private network means better control, security for the network, and the X.25 standard ensures multivendor connectivity, in addition the combination with public networks for dispersed sites with low volume offers the best cost effective implementation.

If your company is willing to optimize or install a wide area network, Hewlett-Packard proposes hybrid networks, which combine HP Private Packet Network and public networks, offering the ultimate flexibility and the best of both worlds.

COMPANY-WIDE NETWORKING



TITLE: OSI Basics and Future Directions

AUTHOR: Bruce Fram

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 2033

DS TO NS MIGRATION ON THE HP 3000

Chris Wallin

HEWLETT-PACKARD

GRENOBLE NETWORKS DIVISION

5, Avenue Raymond Chanas

38320 Eybens, FRANCE

WHAT IS NS?

NS is a family of networking products for the HP 3000 computers. It enables HP 3000 computers to send and receive data from other HP 3000 computers and even from non-HP equipments. NS is also a networking architecture which complies with the OSI (Open Systems Interconnect) seven layer model.

The first member of this family is the NS3000/V Network Services which covers the top most layers of the model. NS 3000/V provides the user with a set of networking capabilities ranging from terminal access (Virtual Terminal) and file transfer (Network File Transfer) to database access (Remote Database Access) and process to process communications (Network Interprocess Communication and Remote Process Management).

Other members of the NS family are all network links which correspond to the lower layers of the model, and which attach the system to the physical network: ThinLAN/V for local area networks, NS Point-to Point/V for synchronous high speed point-to-point connections, Asynchronous Serial Network Link (ASNL) for asynchronous connections over telephone lines, and NS X.25/V for X.25 wide area networks.

WHY MIGRATE TO NS ?

HP's networking offering has included in the past years the DS (Distributed Services) products, and there are many reasons for replacing them by the NS product family.

Broader user capabilities

Users can establish multiple sessions to the same remote system, and transfer groups of files, even between remote systems without the previous establishment of a remote session.

Network Transparency

NS routes data to the destination system, regardless of the number of intermediate systems and the number and type of networks it needs to cross. In consequence, the user does not need to know the network topology and simply enters the destination system name in his commands.

Large Scale Networking

NS is better designed for very large corporate networks which include multiple local or regional subnetworks. NS takes care of the addressing mechanism and allows for great flexibility in network design and implementation.

Compliance to Standard

The NS network links implement industry and de facto standard protocols such as DARPA TCP/IP as the network transport and internet levels for all network links, CCITT X.25, HDLC or LAP-B, as well as IEEE 802.3 for the other levels.

Users can take advantage of these protocols and programmatically access the TCP or X.25 levels to develop special software such as applications for connecting an HP 3000 computer to a non-HP device.

More Applications

NS serves as a platform for future applications and networking products available from HP. For example, users wanting to install NetDelivery on their system or the OpenView network management products will require NS.

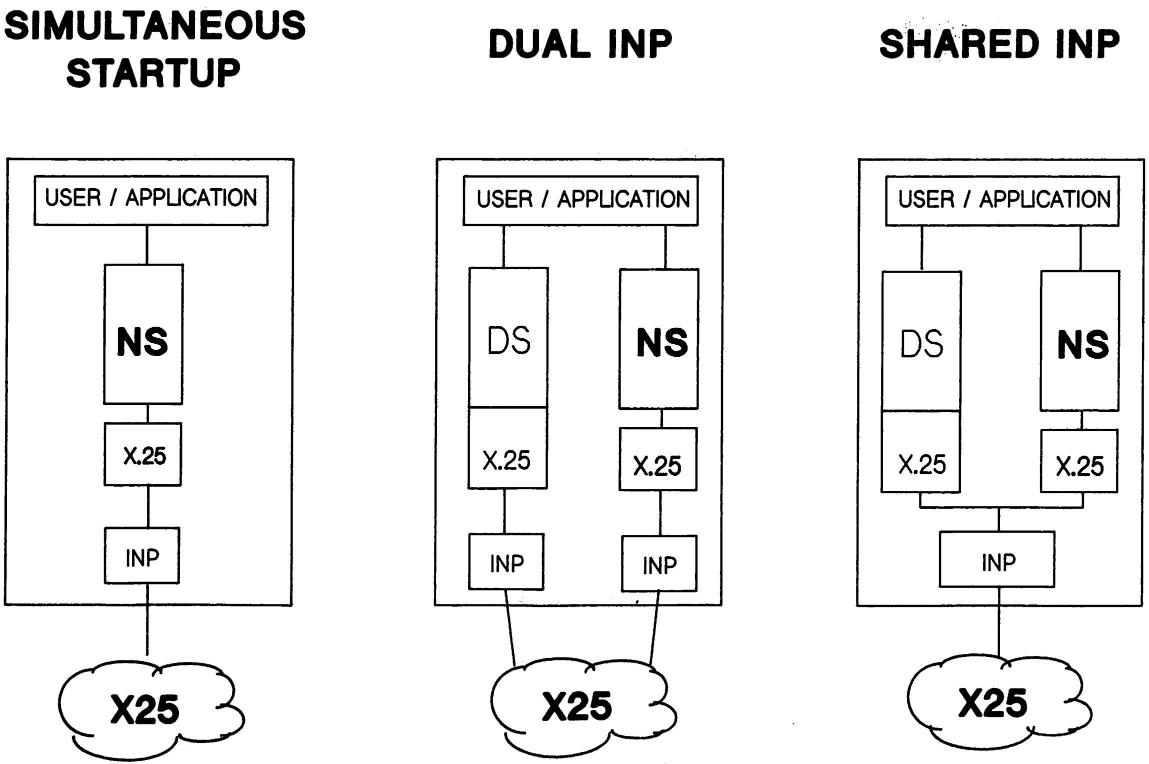
UPGRADE PROGRAM

Special Promotions

HP wants to encourage its customers to migrate their networks from DS to NS and offers special promotions on the NS products. Any customer with support contracts on the products listed hereunder will receive a FREE upgrade to the corresponding NS product:

DS Network Services (32344A/R)	NS 3000/V Network Services (32185A/R)
DS Point-to-Point, Direct (30285A)	NS Point-to-Point 3000/V (30271A)

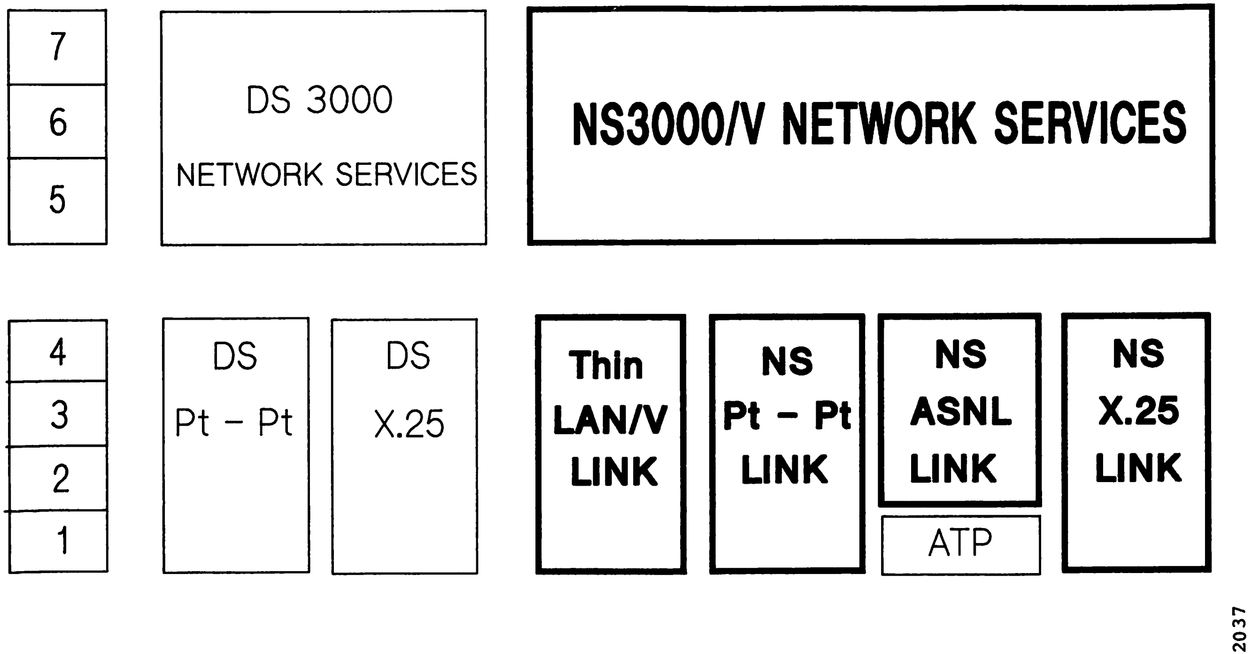
DS-NS MIGRATION STRATEGIES



2037

NS3000/V NETWORK SERVICES & LINKS

PRODUCT STRUCTURE



DS Point-to-Point, Modem (30285A)	NS Point-to-Point 3000/V (30271A)
DS X.25	NS X.25 3000/V (32187A) (24405A)

No Orders

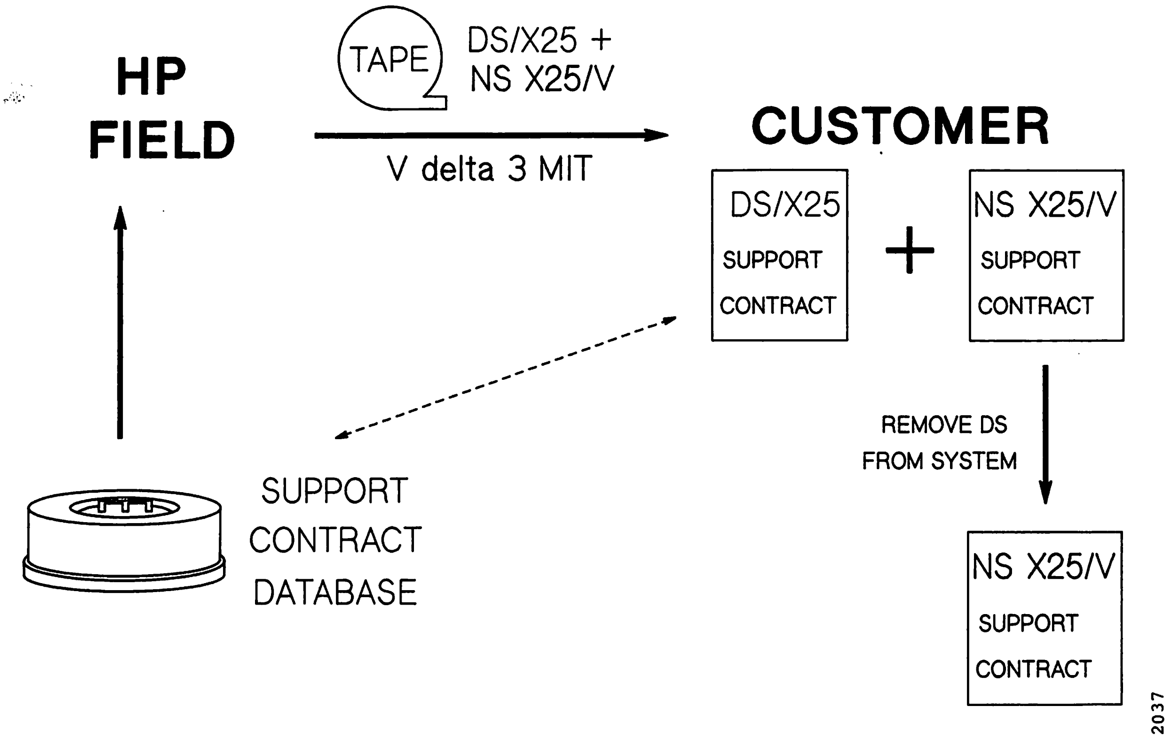
NO ORDERS NEED TO BE PLACED FOR THIS FREE UPGRADE.
To ensure that all DS customers get the opportunity to easily upgrade, the NS software for the four products listed above will be distributed via MIT tapes as part of the normal DS update. Customers simply need to purchase a support contract on the NS product they have received, and they will begin receiving software and manual updates.

The first MIT tape on which this upgrade is available will be V-delta-3, in conjunction with the introduction of NS X.25/V. The free upgrade is planned to be available for a year to ensure a broad coverage of HP's DS customers.

MIGRATION STRATEGIES

There are three primary migration strategies that customers can choose to implement when migrating from DS to NS. The choice of a migration strategy is influenced by a variety of factors such as network topology, type and size, organizational considerations and geographical location of the nodes in the network.

DS TO NS UPGRADE PROCESS



Simultaneous Startup

This is a migration strategy in which the DS network is shut down and the NS network is brought up on all nodes in the network simultaneously. The installation of NS 3000/V links and services can be phased in over a period of time, but it is the actual startup of the NS network that must happen simultaneously. This strategy is appropriate for a small network (less than five nodes) that is geographically centralized and when there is no reason to maintain the DS environment for remote communications to non-NS supported systems such as HP 1000 or S/III, 30,33.

Dual INP Configuration

This strategy requires an additional INP in order to concurrenctly run a DS and NS link. This strategy enables the customer to phase in migration to NS. Consider using this strategy when the network is larger or more complex than the one described above, or when DS connectivity must be maintained. This strategy is appropriate when the network is hierachical in design and most nodes communicate with one central node such as seen in many point-to-point networks.

Shared-INP Configuration

This strategy is supported only for X.25 networks. This requires upgraded DS software and configurations changes to both NS X.25

and DS X.25 in order for them to share an INP. This strategy allows the customer to gradually migrate their network from DS X.25 to NS X.25 and maintain full connectivity of the network throughout the process.

Shared-INP Principles

The shared-INP migration strategy is a phased process which requires some preparation and organization.

The main steps of the process are:

- PREPARATION: Design the network with sub-networks and gateways, examine IP addressing and node naming, verify DS applications will remain unchanged with NS, define access security and back-up routes.

- INSTALLATION & CONFIGURATION: Install all DS and NS software, modify the I/O configuration, enter all NS X.25/V parameters, verify configuration.

- NETWORK DIRECTORY: Create a network-wide network directory and disable all node addresses that still do not have NS installed.

- START-UP: Activate NS in the shared-INP mode and distribute the Network Directory via DS to the remote nodes and activate their NS software modules.

- ADDING NS NODES: As new nodes start-up NS, enable their addresses in the Network Directory.

- MONITORING: When NS is installed on all nodes, ensure all user and application sessions are moving from DS to NS.

- REMOVE DS: When no more network activity is generated with DS, remove it.

The shared-INP strategy will be illustrated during the presentation by a four node network migration example.

MIGRATION TOOLS AND SERVICES

There are a variety of Network Support tools available to help customers plan for migration.

The Network Planning and Design service is used to conduct an in-depth needs analysis to develop a customized and comprehensive network design. This can be used to develop a detailed DS to NS migration plan. Network Prepare can be purchased to provide assistance in developing an implementation schedule for migration.

HP-to-IBM Communications

Michael Strickland

Hewlett Packard Company

19420 Homestead Rd.

Cupertino, CA 95014

IBM connectivity is an important element of Hewlett-Packard's HP AdvanceNet strategy for multivendor networking, and the company offers a full line of products that integrate HP computers with IBM systems. These products allow subnetworks of distributed systems to coexist with IBM systems at corporate or divisional headquarter locations.

A strong commitment to multivendor networking based on international and de facto standards, such as SNA, is central to the HP AdvanceNet strategy.

HP provides batch and interactive SNA communications on the HP3000 family of business computers in both gateway and standalone configurations.

The HP3000 batch communications product, SNA/NRJE, enables an HP3000 to appear as an 8100 DPPX/RJE Workstation to an IBM host, when used with HP's SNA Link product. Batch jobs are spooled when communications lines are busy, and automatically processed when the lines are free.

HP's SNA Interactive Mainframe Facility (IMF) is the company's interactive communications product. SNA IMF emulates the main features of an IBM 3270 control unit, using SNA PU2 and LU1, 2 and 3 protocols. HP terminals, printers, and applications on the HP3000 that are running SNA/IMF emulate IBM terminal and printer functions.

SNALink/3270 on the HP Vectra PC and the HP Portable Plus PC provides interactive access to IBM 3270 applications by emulating an IBM 3274 control unit with 3278 display stations and 3287 printers attached.

For program-to-program applications between an HP3000 and an IBM host, users can either use the 3270 datastream based programmatic interface in SNA IMF or the LU6.2 API product. The LU6.2 Application Programming Interface (API) enables third parties or end users to develop HP3000 applications that use the LU 6.2 protocol to communicate with IBM systems.

HP Information Access and the Information Access Cullinet Link provide a controlled way to link PC users with mainframe databases. The PC user can transfer files in Lotus 1-2-3, R Base:5000, dBase II, or other popular PC word processing and graphics applications to or from Cullinet C/ICMS tables on an IBM mainframe.

Thus users can enter transactions on the HP3000, send information to the host's central database, and retrieve information from that database.

HP additionally offers bisynchronous (BSC) HP-to-IBM products, including: 2780/3780 (RJE) batch workstation emulation, HASP workstation emulation, and 3270 terminal emulation.

Office Connections

Hewlett-Packard offers connections to IBM's PROFS and DISOSS, as well as support of IBM's Document Content Architecture (DCA), Document Interchange Architecture (DIA), and Logical Unit Type 6.2 (LU 6.2).

HP OfficeConnect to DISOSS enables HP DeskManager users to exchange electronic mail with DISOSS users, and to file, search for, retrieve and delete documents in a DISOSS library. The product also includes a converter that permits exchange of documents with DISOSS in Final Form Text DCA.

HP OfficeConnect to PROFS allows users of IBM's PROFS and HP3000's to communicate through their own electronic mail systems.

HP OfficeConnect products provide HPDesk users with transparent access to PROFS and DISOSS. No HP software is required for the IBM system.

Hewlett-Packard has verified the performance of third-party solutions for Reverse Pass-Through and SNA-to-X.25 conversion. Reverse pass-thru makes the IBM 3270-to-HP terminal conversion needed to allow users to access applications based on the HP3000.

SNA/X.25 conversion allows the SNA HP-to-IBM communication products to be used over an X.25 network.

Hewlett-Packard currently has an entire R & D facility working on the development of new products that enable users to communicate within IBM environments; among these products are:

-- HP's software-only, enhanced reverse pass-through product;

-- SNA-to-X.25 conversion software.

Hewlett-Packard's objective is to make HP StarLAN, which is the company's strategic commercial LAN product for the office, useful in the widest possible range of office environments.

To meet this goal, Hewlett-Packard has tested HP StarLAN over IBM Cabling Types 2 and 3 (shielded and unshielded twisted pair wiring, which is used by both StarLAN and Token Ring). HP will run tests with other wiring (such as IBM Cabling Plan Types 1 and 9) as customer need requires.

HP firmly believes that HP StarLAN offers several advantages over Token Ring, but we are also commited to protecting customers' investments in our own office solutions as well as those of IBM. To this end, HP plans to enable Vectra PCs that are linked to Token Ring networks to access Personal Productivity Center (PPC) functionality on the HP3000 through an 802.3/802.5 bridge. HP's Vectra PC third party program will also include a Token Ring adapter card and software. In addition, HP plans to offer a method of connection for HP 802.3 LAN attached Vectra PCs to gain access to an IBM mainframe on the Token Ring Network.

The advantages which StarLAN offers to customers include a lower cost per connection than Token Ring. In addition, since HP StarLAN is designed for use over phone wire, customers can feel confident that StarLAN will work with fewer limitations and permit more flexible configurations over existing phone wire (which is unshielded twisted pair).

HP is commited to continually enhance our capabilities in SNA and SNA-based architectures, so we have made SOI's for IBM Network Management Architecture and PU 2.1.

To provide coexistance between HP industry-standard networking an IBM's SNA networking, HP will support IBM's Network Management Architecture (NMA) on the company's SNA HP-to-IBM products. A distinction, however, should be made between NMA and NetView. NetView is an IBM product packaging and not an architecture. HP is investigating how it will implement support of NMA.

This does not necessarily imply that HP will support NetView/PC. Given the limited functionality currently defined in NetView PC coupled with the present limitations of the PC itself, we are not convinced that NetView PC will be able to provide our customers with the type of network management capability they need. We believe that our recently introduced Private Packet Network (PPN) offerings can deliver more state-of-the-art functionality than NetView/PC offers.

At present, HP has developed a network management architecture that parallels as closely as possible future ISO standards for network management. HP's architectured solution will provide applications that span all HP processors and multivendor environments.

On June 16, 1987, IBM for the first time, announced their intent to support Node Type 2.1 on their strategic processor line, the System 370 for delivery in 3rd Quarter 1988. Accordingly, peer-to-peer connectivity is still an emerging technology for IBM. In HP's continuing commitment to provide means for HP networks to coexist with IBM, HP now formally commits to provide a Node Type 2.1 for the HP3000.

A number of connectivity options will arise as IBM begins to deliver Node Type 2.1. Examples are: LU 6.2 Application Programming Interface, other LU 6.2-based applications and peer-to-peer communications between two Node Type 2.1 peripheral nodes over an SNA backbone network. HP will assess the need to these capabilties as they apply to our customers and to the HP3000.

HP also offers a number of products for HP9000 to IBM communcations. For batch communications, SNA users can use HP-UX SNA3770 to exchange files between the HP9000 and an IBM host ; Series 300 BSC users can use BSC RJE to perform the same function.

For interactive communications, HP-UX SNA3270 provides 3274 emulation which allows HP9000 terminals and printers to emulate IBM 3278 terminals and 3287 printers. In addition, HP-UX SNA3270 supports IBM 3270 PC File Transfer which makes it possible to exchange files with the IBM host with a one line command.

HP-UX SNA3270 and HP-UX SNA3770 can be accessed in a standalone or gateway configuration. HP-UX SNA3270 and HP-UX SNA3770 can also share the same link to a host.

In summary, Hewlett-Packard is firmly commited to making connectivity to IBM easy to implement and use for its customers. This is an important element of HP's commitment to multivendor networking based on both international and de facto industry standards.

HP ADVANCENET STRATEGY

* **PROVIDE FOUNDATION FOR HP'S COMPUTER PLATFORMS**
  - PEER-TO-PEER COMMUNICATIONS OVER LAN AND WAN
* **PRACTICAL APPLICATION OF INDUSTRY/DE FACTO STANDARDS TO SOLVE MULTIVENDOR NETWORKING NEEDS**
  - HP-HP & MULTIVENDOR: OSI
    - SHORT TERM: OSI + ARPA
    - LONG TERM: FULL OSI (MAP, TOP, ISDN)
    - OTHER DE FACTO STANDARDS
  - HP TO IBM: SNA
* **QUALITY PRODUCTS AND SERVICES**
  - FUNCTIONALITY, LOCALIZABILITY, USABILITY, RELIABILITY, PERFORMANCE, SUPPORTABILITY
  - CONSULTING, SUPERVISION, AND SUPPORT
* **AREAS OF EMPHASIS**
  - INFORMATION ACCESS
  - NETWORK MANAGEMENT AND REMOTE OPERATOR SUPPORT

Information Networks Division
IBM00

HEWLETT PACKARD

* A strong commitment to multivendor networking based on international and de facto standards, such as SNA, is central to the HP AdvanceNet strategy.

* HP currently has an entire R & D facility working on the development of new products that enable users to communicate within IBM environments.

HP–IBM NETWORKING

CUSTOMER NEED:

* *ABILITY TO COMMUNICATE WITH INSTALLED IBM SYSTEMS AND NETWORKS.*

HP'S OBJECTIVE:

* *QUALITY PROVIDER OF MULTIVENDOR NETWORKING USING INDUSTRY STANDARDS & DEFACTO STANDARDS SUCH AS SNA.*

HP'S OFFERINGS:

* *HP–IBM COMMUNICATION PRODUCTS IN 3 ENVIRONMENTS:*
  - *BUSINESS OFFICE*
  - *MANUFACTURING*
  - *ENGINEERING & RESEARCH*
* *CHOICE OF GATEWAY OR STANDALONE IMPLEMENTATION.*

Information Networks Division
IBM01

HEWLETT PACKARD

* HP offers a full line of products that integrate HP computers in different solutions with IBM systems.

* These products allow subnetworks of HP distributed systems or standalone systems to coexist with IBM systems at corporate or divisional headquarters locations.

HP-IBM NETWORKING

WORK ENVIRONMENTS

	BUSINESS OFFICE, REG. SALES & SERV. & MFG. DATA CENTER HP3000 & PC	MFG. REAL-TIME HP1000	BUSINESS OFFICE ENGINEERING & CIM HP9000
BATCH (RJE) COMMUNICATIONS	SNA NRJE RJE MRJE SNA Server	RJE MRJE	HP-UX SNA 3770 (SERIES 300, 800 UNIX)* RJE EMULATOR (SERIES 200, 300 500 UNIX)
INTERACTIVE (3270) COMMUNICATIONS	SNA IMF & BSC IMF SNA 3270 LINKS (VECTRA & PORTABLE +) 3278 IRMA (VECTRA & TOUCHSCREEN) SNA Server	PMF (PROGAMMATIC)	HP-UX SNA 3270 (SERIES 300, 800 UNIX) 3270 EMULATOR (SERIES 200 + 300 BASIC & PASCAL)
OFFICE CONNECTION	OFFICE CONNECT TO DISOSS & LU6.2 BASE OFFICE CONNECT TO PROFS HP CONVERT/DCA INFO ACCESS		

* Future Product

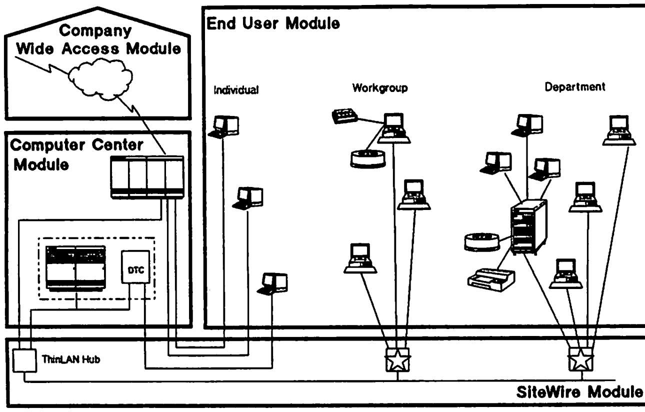
Information Networks Division
IBM02

HEWLETT PACKARD

* For the HP3000 family of business computers, HP supports a number of HP-to-IBM products that offer batch, interactive, program-to-program, and office connection functionality.

* After a quick overview of how HP to IBM communications fits into the Business Office solution, we will look at these products for the HP3000.

Business Office Solution Summary

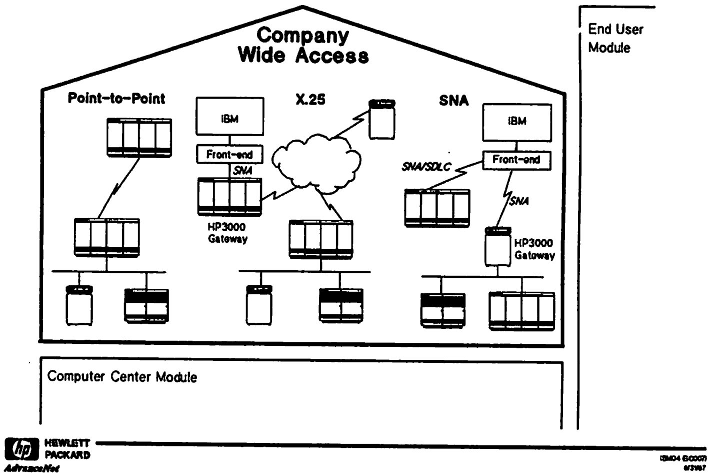


HEWLETT PACKARD
AdvanceNet

* In the Business Office there is sometimes the need to send accounting information from a remote office to a corporate mainframe, or to submit payroll figures to an IBM host.

* HP AdvanceNet offers high-performance packages that maximize productivity by making tasks like these fast and easy.

Company Wide Access Module



* There are several ways to connect HP3000's to IBM mainframes.

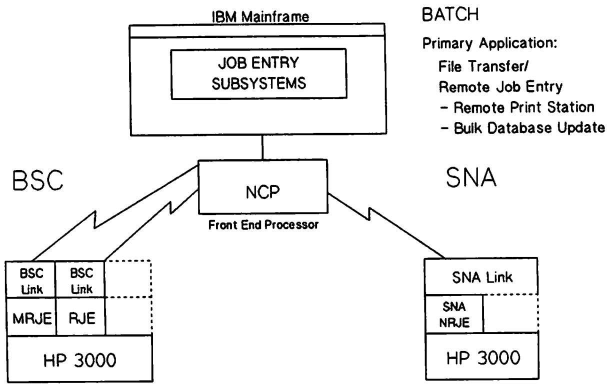
* If the HP3000 is not on a LAN, a standalone connection can be used to connect directly to an IBM host.

* If several HP3000's on a LAN require SNA access, one of these systems can serve as a non-dedicated SNA gateway.

* The SNA gateway can be locally attached to the IBM host and accessed by remote locations through an X.25 network, or the SNA gateway can can be locally attached to the LAN with a remote connection to the IBM host.

HP - IBM NETWORKING:

Business Office Environment



Information Networks Division

HEWLETT PACKARD

* Starting first in the batch area, we'll now take a look at some of the services which are available.

* Batch products can be used as an easy way to exchange files with an IBM host. Batch jobs are spooled when communications lines are busy, and automatically processed when the lines are free.

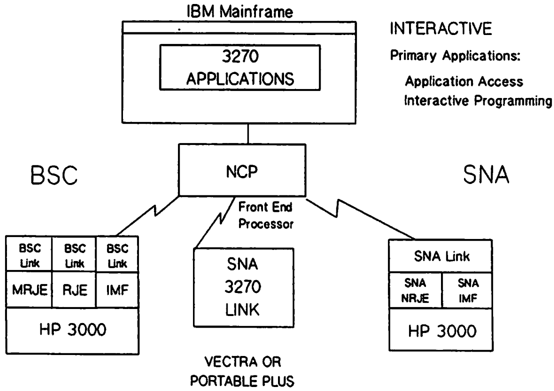
* Batch products can also be used to subit jobs to a host, or for print jobs, or for bulk database update.

* SNA users can use SNA/NRJE which enables an HP3000 to appear as an 8100/DPPX workstation to an IBM host, when used with the SNA Link product.

* For BSC users, RJE provides 2780/3780 emulation, while MRJE provides HASP emulation.

HP - IBM NETWORKING:

Business Office Environment



Information Networks Division

IBM06

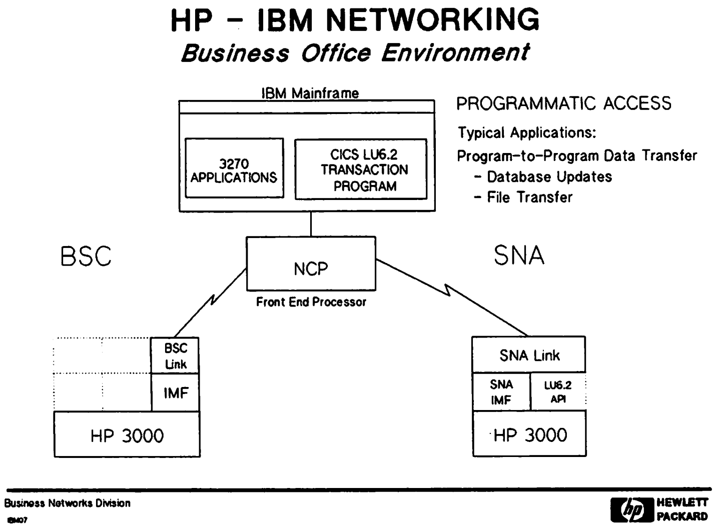
HEWLETT PACKARD

* In the interactive area, users can access applications on the host and do some local printing. HP terminals and printers are allowed to emulate IBM terminal and printer functions.

* SNA Interactive Mainframe Facility (IMF) emulates the main features of an IBM 3270 control unit, using SNA PU2 and LU1, 2, and 3 protocols.

* For BSC networks, IMF offers similar functionality.

* The SNALink/3270 product for the HP Vectra PC and the HP Portable Plus PC provides interactive access to IBM 3270 applications by providing PU2 emulation.



* Both IMF and SNA IMF also provide a 3270 applications programming interface. A program running on the HP3000 can access information from a program running on the IBM host, and vice-versa.

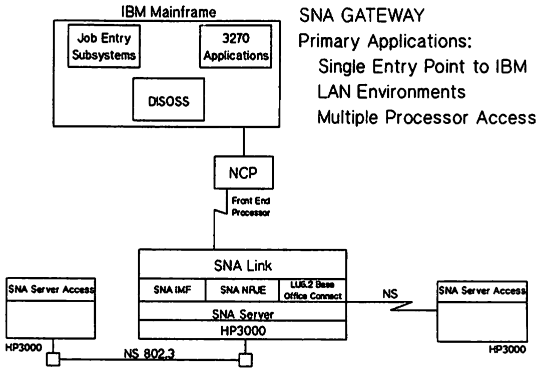
* This capability can be used for database access and updates, or for file transfer.

* Another alternative to the 3270 programmatic interface is to use the LU6.2 API product.

* While the LU6.2 sessions to hosts early in 1988 were restricted to CICS access, VTAM/APPC will provide much better functionality when it is released at the end of this year. VTAM/ APPC will allow applications running in IMS, TSO, and CMS to access LU6.2 sessions.

HP – IBM NETWORKING

Business Office Environment



Information Networks Division

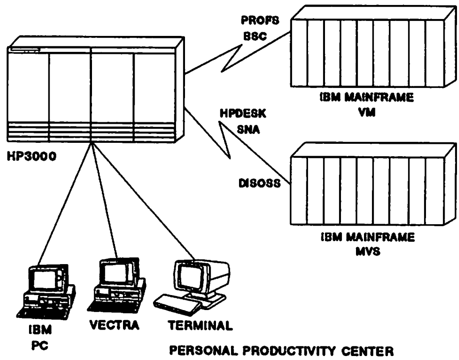
HEWLETT PACKARD

* While we have been showing standalone connections up until now, users can also access SNA sessions to the IBM host through the SNA Server gateway.

* Other nodes on the LAN or linked remotely through a point-to-point link can access SNA services as if there were an SNA Link on that system.

FULLY INTEGRATED HPDESK SOLUTIONS TO DISOSS OR PROFS

HP OFFICECONNECT TO DISOSS AND PROFS



Information Networks Division

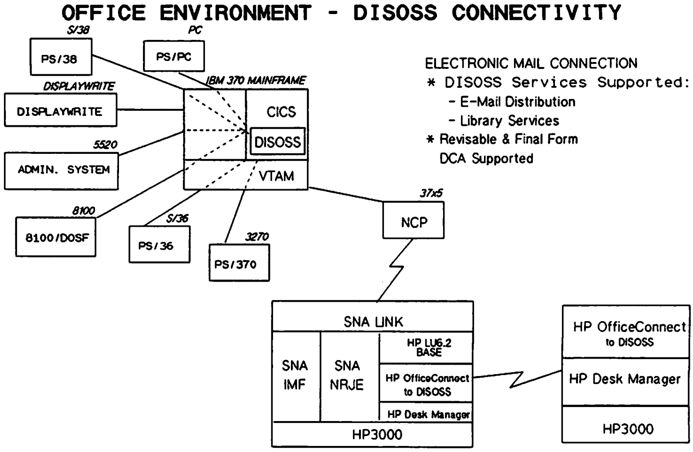
HEWLETT PACKARD

* Two office connect products allow HPDesk users to exchange mail with PROFS and DISOSS users.

* For VM users, HP Officeconnect to PROFS allows HPDESK mail to be exchanged with VM through a BSC link. (BSC RJE is also required.)

* For MVS users, HP Officeconnect to DISOSS allows the exchange of mail between HPDESK and DISOSS through a more complicated SNA link.

HP-IBM NETWORKING
OFFICE ENVIRONMENT - DISOSS CONNECTIVITY



Information Networks Division

HEWLETT PACKARD

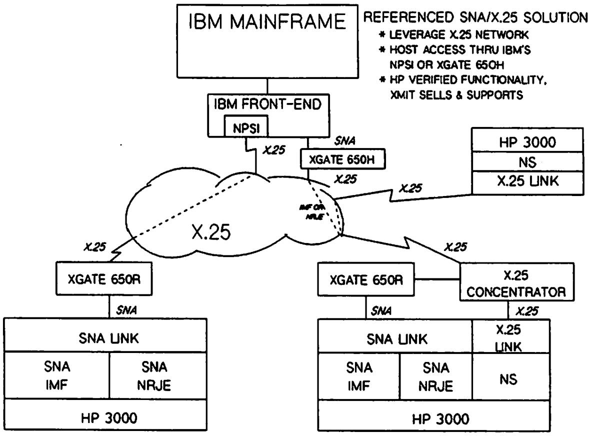
* HP OfficeConnect to DISOSS enables HP DeskManager users to exchange electronic mail with DISOSS users, and to file, search for, retrieve and delete documents in a DISOSS library.

* HPOfficeconnect to DISOSS emulates some of the functions of a Displaywriter. This provides a DIA interface to the host over an LU6.2 session. This product also supports the Final Form DCA format.

* The user has the option of converting HP3000 documents to revisable form DCA format with the HP Convert/DCA product.

* Although the technology used here is quite impressive, most IBM users have not adopted DISOSS. DISOSS appears to consume a large amount of resources and has a complicated user interface.

SNA OVER X.25 VIA XMIT



Information Networks Division

HEWLETT PACKARD

* With IBM's recent release of NPSI, the performance of SNA over X.25 is much improved on the FEP end -- on the other end, a referenced product from XMIT called XGATE 650R lets the HP3000 handle the SNA over X.25 traffic.

* The HP3000 on the right is using the X.25 network to access both an HP3000 and an IBM host.

3rd PARTY PRODUCT

REVERSE PASS THRU

* HP Recommended Alternatives

- 3rd Party
  Low End: COAX 3270 from Gandalf and
  PREVIEW from Tymlabs
  High Performance: Gateway/1000 from Forest Computer

- Line Mode, HPDESK Access Only
  Network Project Center Special

Information Networks Division

HEWLETT PACKARD

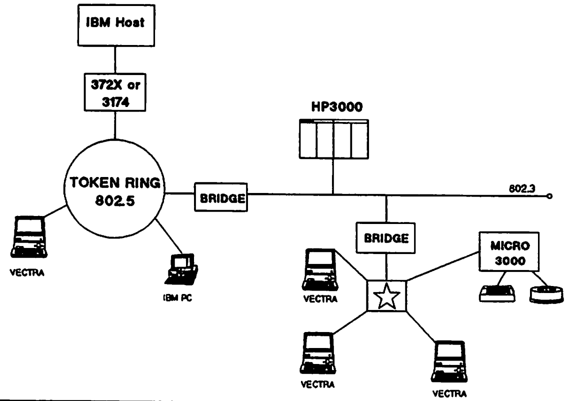
* For IBM host users who would like interactive access to an HP3000, there are several solutions which are available today.

* At the low end, COAX 3270 from Gandolf converts ASCII data to EBCDIC data and vice-versa, giving the user line mode access. By running PREVIEW on the HP3000, the 3278 terminal can access V-Plus block mode applications. Each 3278 requires a terminal port on the HP3000.

* A higher performance solution called Gateway/1000 uses a DHCF connection to the host and some terminal ports on the HP3000. This solution provides V-Plus block mode access but without extended attributes. Any 3278 attached to the IBM host can access an HP3000 session.

* For customers needing line mode access to HPDESK only, there is a special which is available from the HP ING Network Project Center.

UNDER DEVELOPMENT:
STARLAN AND TOKEN RING COEXISTENCE



* (0) HP has run tests that show that StarLAN can be run over IBM Cabling Plan Types 2 and 3.

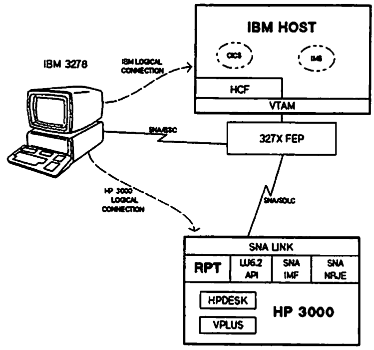
* (1) IBM's Token Ring card will be used in Vectra PC's.

* (2) A Token Ring to 802.3 bridge is being done. There should be no throughput limitation, since it is a fast bridge.

* (3) -PC's on Token Ring can access other PC's using IBM PC LAN program.
  -PC's on Token Ring can access the IBM mainframe through the IBM 3270 gateway (IBM NetBIOS), IBM PU2.0 emulation S/W (SNA over TR), or a DCA 3270 gateway (HP NetBIOS).
  -PC's on Token Ring can access the HP3000 through the bridge, using HP NetBIOS over TCP/IP.

* (4) PC's on 802.3 access the IBM mainframe through the bridge to a DCA gateway (HP NetBIOS).

UNDER DEVELOPMENT: REVERSE PASSTHRU
(BUSINESS OFFICE & REG. SALES AND SERVICE SOLUTIONS)



PRIMARY USAGE:

IBM 3270 TERMINAL ACCESS TO

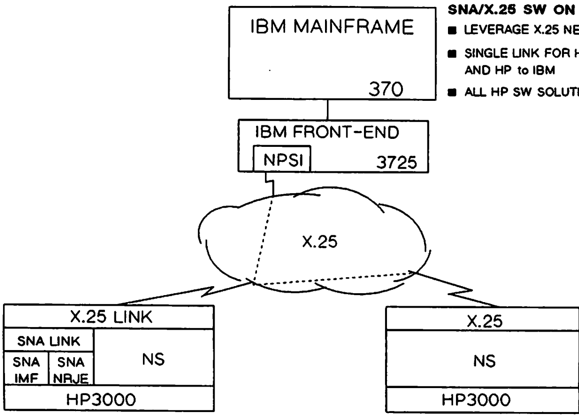
- HPDESK
- MM AND PM
- OTHER VPLUS APPLICATIONS
- TTY APPLICATIONS

Information Networks Division
IBM17

HEWLETT PACKARD

* A reverse passthru S/W product is under development which would not require any dedicated HP3000 terminal ports. Through the IBM Host Command Facility (HCF), IBM users will be able to access V-plus block mode applications.

UNDER DEVELOPMENT: SNA OVER X.25



SNA/X.25 SW ON HP3000

- LEVERAGE X.25 NETWORK
- SINGLE LINK FOR HP to HP AND HP to IBM
- ALL HP SW SOLUTION

* Also under development is a product which would allow SNA over X.25 from an HP3000. An HP3000 at a single node address would be able to communicate with HP3000's and IBM mainframes.

FUTURE DIRECTIONS

* IBM NETWORK MANAGEMENT ARCHITECTURE SUPPORT

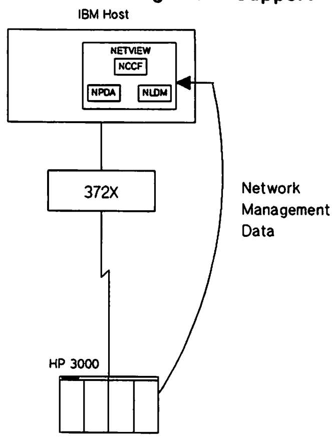
* PU2.1 (NODE TYPE 2.1) SUPPORT

Information Networks Division

HEWLETT PACKARD

* HP is committed to continually enhance capabilities in SNA and SNA-based architectures, so we have made SOI's for IBM's Network Management Architecture and PU2.1.

FUTURE DIRECTION:
IBM Network Management Support



Information Networks Division

HEWLETT PACKARD

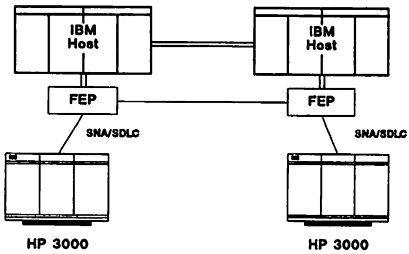
* To provide coexistance between HP industry-standard networking and IBM's SNA networking, HP will support IBM's Network Management Architecture (NMA) on the company's SNA HP-to-IBM products. A distinction, however, should be made between NMA and NetView. NetView is an IBM product packaging and not an architecture. HP is investigating how it will implement support of NMA.

* HP will forward to Netview the alerts which are supported by the emulated subsystem. For example, SNA IMF emulates a 3274 cluster controller, so SNA IMF will forward the alerts to Netview which are supported by the 3274.

IF ASKED,

* This does not necessarily imply that HP will support NetView/PC. Given the limited functionality currently defined in NetView PC coupled with the present limitations of the PC itself, we are not convinced that NetView PC will be able to provide our customers with the type of network management capability they need.

Under Development: LU6.2/PU2.1 SUPPORT



-Parallel sessions
-Multiple sessions

Information Networks Division
IBM21

HEWLETT PACKARD

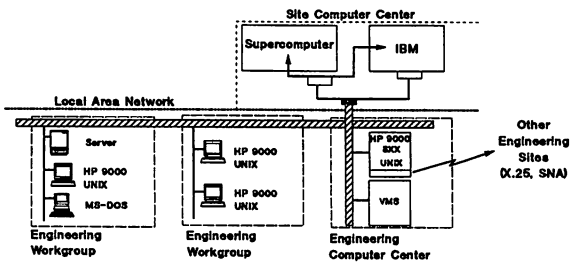
* With IBM's NCP support for PU2.1 at the end of 1988, it will be possible to route LU6.2 sessions through an FEP. This would allow the two HP3000's to have an LU6.2 session through the two FEP's.

* PU2.1 support for the HP3000 will also allow parallel and multiple sessions. Parallel sessions allow several applications to share one LU6.2 to LU6.2 session to another system. Multiple sessions allows one LU6.2 to talk to multiple LU6.2 sessions that may be on the same and/or different hosts.

  FOR LARGE PUBLIC AUDIENCES:

* Other services are under investigation.

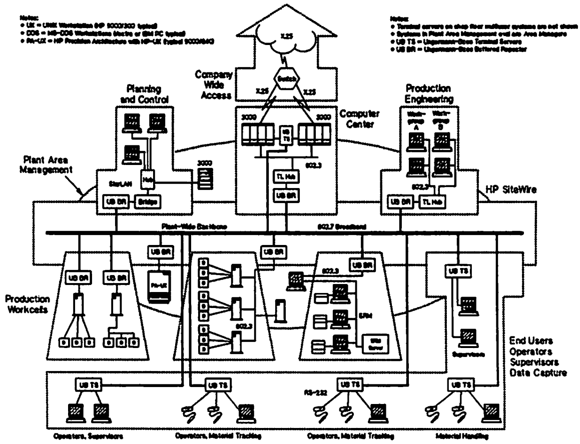
HP AdvanceNet for Engineering



* We'll now take a look at our HP-UX to IBM products. HP-UX is used in the HP9000 in the Business Office, Engineering, and CIM solutions. We can skip the Business Office since we have previously covered it for the HP3000.

* In the Engineering solution, engineers and designers need to get higher quality products to market faster. HPAdvanceNet enables an HP9000 user to access applications, data and peripherals on the IBM host.

HP AdvanceNet for CIM: Putting It All Together

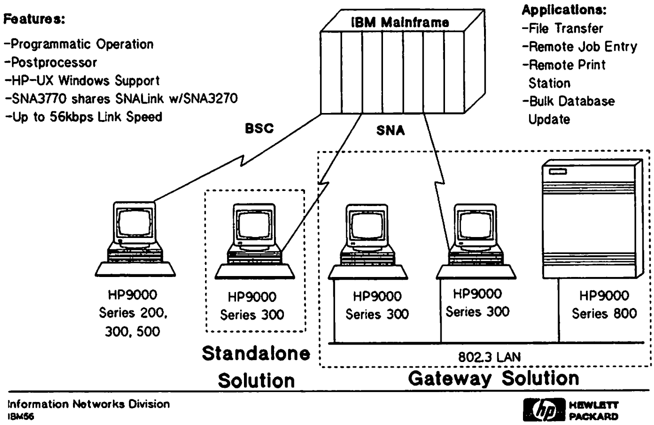


Information Networks Group
IBM55 (CIM22 BC 9/14/87)

HEWLETT PACKARD

* In the CIM solution, there may be the need to access an IBM host from the Computer Center. Or in some cases, there may be an IBM computer in the Computer Center which needs to download information to area managers or cell controllers on the factory floor.

HP-UX to IBM BATCH NETWORKING



* Batch products can be used as an easy way to exchange files with an IBM host. Batch jobs are spooled when communications lines are busy, and automatically processed when the lines are free.

* Batch products can also be used to subit jobs to a host, or for print jobs, or for bulk database update.

* SNA users can use SNA3770 which enables an HP9000 to appear as a 3777 batch workstation to an IBM host, when used with the SNA Link product.

* The product can be used programmatically from Unix. This feature might be used to perform batch updates late at night when computer rates are less expensive.

* The product also has a post processor capability, which allows the user to specify a program which can be started when the job output returns.

* For the S300, there is a BSC RJE product to the host.

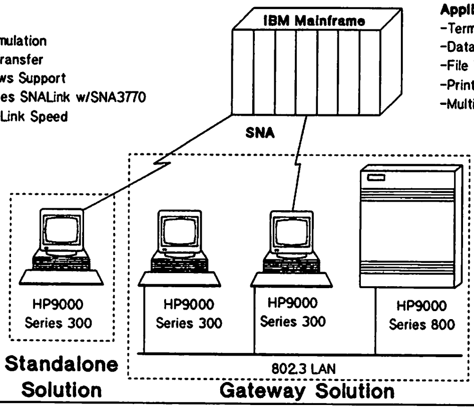
HP-UX to IBM INTERACTIVE: SNA3270

Features:

-3278, 3287 Emulation
-PC3270 File Transfer
-HP-UX Windows Support
-SNA3270 shares SNALink w/SNA3770
-Up to 56kbps Link Speed

Applications:

-Terminal Access
-Database Access
-File Transfer
-Printing
-Multiple Sessions



Information Networks Division
IBM37

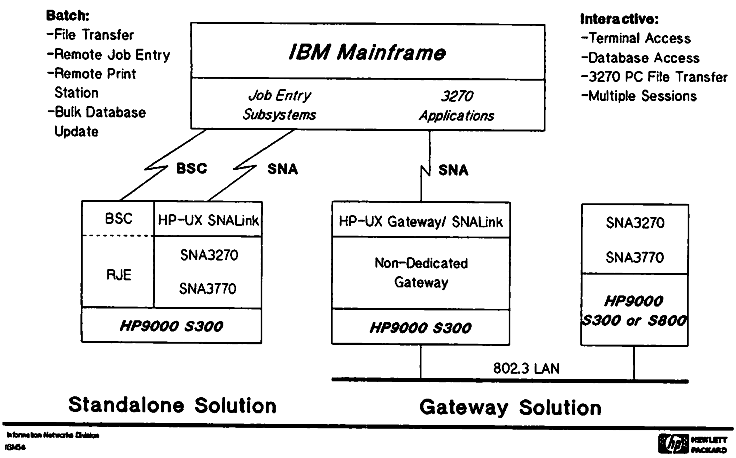
HEWLETT PACKARD

* In the interactive area, users can access applications on the host do some local printing, and transfer files. HP9000 attached terminals and printers are allowed to emulate IBM terminal and printer functions.

* HP-UX SNA3270 emulates the main features of an IBM 3274 control unit, using SNA PU2 and LU1, 2, and 3 protocols.

* HP-UX SNA3270 also supports PC3270 file transfer to and from the IBM host. Files can be transfered to and from the host with a one line command! (This capability requires that the host have the PC3270 program available.)

COMPLETE HP-UX to IBM NETWORKING:



* One SNA Link can be used to support both batch and interactive communications to the host.

HP-IBM NETWORKING ADVANTAGES

BUSINESS SYSTEMS

* DISOSS CONNECTION INTEGRATION WITH HP DESK MANAGER
* PROFS CONNECTION TO 9370 DEPARTMENTAL SYSTEMS
* CHOICE OF GATEWAY OR STANDALONE
* REVERSE JOB ENTRY: IBM —> HP
* SOPHISTICATED JOB OUTPUT MANAGEMENT

ENGINEERING SYSTEMS

* HIGH PERFORMANCE UNIX TO SNA 3270 GATEWAY
* IBM CONNECTIONS FOR UNIX, BASIC, AND PASCAL ENVIRONMENTS

PC'S

* CHOICE OF: COAX 3278 EMULATION (FOR VECTRA AND TOUCHSCREEN) OR 3274 PU2 REMOTE SNA CONNECTION (FOR VECTRA AND PC PORTABLE)
* TRANSPARENT ACCESS TO IBM MAINFRAME CULLINET DATABASE

Information Networks Division
IBM80

HEWLETT PACKARD

* Summary slide.

HP-IBM NETWORKING

SUMMARY

* COMPLETE SNA FAMILY OF BATCH, INTERACTIVE, PROGRAM-TO-PROGRAM AND OFFICE HP-IBM DATA COMMUNICATIONS PRODUCTS

* CONNECTIVITY TO IBM IN BUSINESS OFFICE, REGIONAL SALES & SERVICE, MANUFACTURING/CIM AND ENGINEERING ENVIRONMENTS

* COMMITMENTS TO CONTINUALLY ENHANCE CAPABILITIES IN SNA AND SNA-BASED ARCHITECTURES WHILE IMPLEMENTING INTERNATIONAL STANDARDS IN HP ADVANCENET PRODUCTS

Information Networks Division
IBM81

HEWLETT PACKARD

* Summary slide.

HP StarLAN Networking

Alexa F. Ford

Hewlett-Packard Co., Inc.

Colorado Networks Division

3404 E. Harmony Rd.

Ft. Collins, Colorado 80525

Introduction

Hewlett-Packard has received considerable attention over the past few months after announcing a 10 Mbps LAN that runs over unshielded twisted-pair wire: HP StarLAN 10. In addition, HP has been recognized for its leadership role in developing an IEEE standard for this technology. This paper discusses the two networking solutions available from HP based on a star topology and unshielded twisted pair wiring: HP StarLAN and HP StarLAN 10.

HP's site wiring architecture follows a distributed star topology which is compatible with existing telecommunication systems. HP StarLAN is a low cost, 1 Mbps network solution. HP StarLAN 10 provides a 10 Mbps networking solution for environments where heavy network traffic and use of high performance workstations require a wider bandwidth. When combined with the OfficeShare Family of Network Software and Business System Plus, users can take advantage of shared data and resources and the best PC/Mini integration in the industry. [1]

This paper includes information about how these networking links can be used to meet organizational needs providing users cost effective access to information and resources. It will also discuss how HP's commitment to standards has influenced these products and review some configuration information on HP StarLAN, HP StarLAN 10 and mixed cable networks. This paper is organized into the following sections:

- Topology and Wiring
  - Topology Advantages
  - HP SiteWire Architecture
- Meeting Organizational Needs
  - HP StarLAN and StarLAN 10 capabilities
  - Advantages of the HP Solution
  - Scalability of HP Networks
  - Performance
  - Commitment to Standards
- HP StarLAN, HP StarLAN 10 and Mixed Cable Networks
  - HP StarLAN
  - HP StarLAN 10
  - Mixed Cable Networks
- Summary

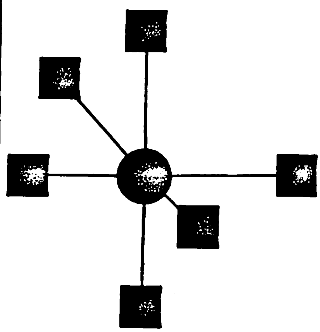
Topology and Wiring

The wiring infrastructure of a building or campus is the foundation upon which any well conceived network design must rest. If this foundation is inadequate it is impossible to build an effective, flexible and manageable network solution.

While the cost of network user interface hardware and software has been decreasing significantly, the cost of network cabling has remained relatively constant. In addition, building wire has a life span two to six times greater than the equipment it connects, increasing the strategic importance of wiring. Because of this, the choice of wiring media is one of the most important long-term decisions management can make.

To aid in this decision, HP has developed a complete set of communications wiring guidelines, products and services, called HP SiteWire. HP SiteWire adheres to an open, multi-vendor wiring foundation and is based on emerging industry standards. For both StarLAN and StarLAN 10, HP's site wiring architecture follows a distributed star topology compatible with existing telecommunication systems.

Advantages of a Distributed Star Topology



- Easy to add, move and make changes
- Reduces cost of network user moves and changes
- Easier network cable administration
- Minimizes costs associated with network downtime
- Mean time to repair is typically less than with other topologies
- Isolation of failures
- Better security

Diagram 1

Advantages of a Distributed Star Topology

As mentioned, HP StarLAN and HP StarLAN 10 use a distributed star topology. A star tolopogy connects individual nodes to a central device which performs all the logic and switching functions required of the system. Star topologies provide a number of benefits including the following items:

Easy to add, move and make changes:

Because wiring closet management using cross connect blocks is centrally managed, network managers are provided with the most simple, expedient method to add or move LAN nodes.

Reduces cost of network user moves and changes:

The cost of user moves and changes can be reduced with a star topology and unshielded twisted pair wire from an average of $1,000 - $1,500 for coaxial cable changes to $200 -$300 per move or change.[2] A savings of about 80%. Most locations are already wired and as a result the cost of moves and changes drops significantly.

Minimizes costs associated with network downtime:

With the auto-segmentation capability of the hub, faults are usually be contained to a single node, rather than impacting the entire network

Mean time to repair is typically less than with other topologies:

Again, troubleshooting is more centralized, with all nodes connecting to a central hub.

Better security can be provided with a Star topology:

Users cannot add themselves to the network without accessing a port on the Hub. The Hub is typically located in the wiring closet which can be locked.

HP SiteWire provides a wiring infrastructure that will meet the requirements for voice, data and office products. The same wiring system that supported only the phone system can now include support for users' data and voice applications.

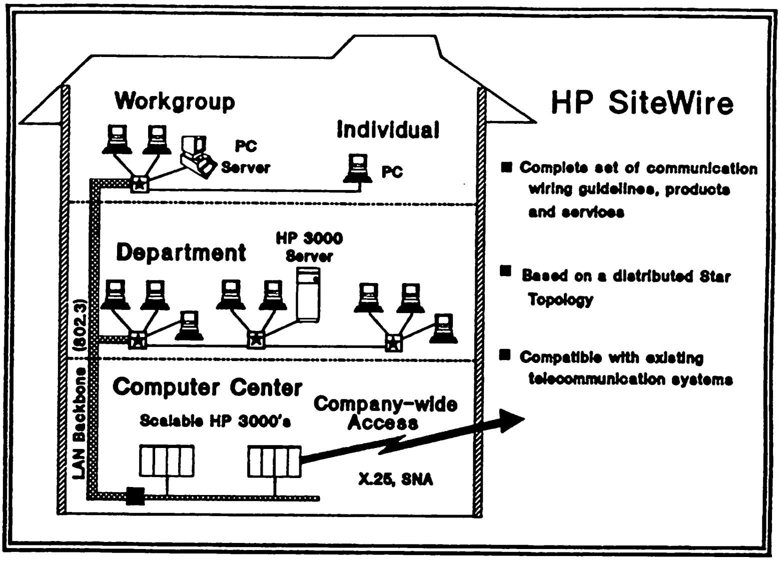


Diagram 2

HP SiteWire Architecture

HP SiteWire Architecture

HP SiteWire architecture uses a thin or thick coaxial backbone cable running horizontally and/or vertically through a building, connecting horizontal subnets of unshielded twisted pair or coaxial cabling, depending on the environment.

There are numerous benefits to using of unshielded twisted pair wiring in the work area compared to shielded twisted pair wiring or coaxial cable.

Lower cost and ease of installation:

Unshielded twisted pair cable costs less and is easier to install. The wire requires less physical space than coax or shielded twisted pair wire, making installation easier and requiring less space in ducts and satellite closets.

Flexibility:

Unshielded twisted pair wire is more pliable than coaxial or shielded twisted pair wire. The pliability of the wire makes installation of the wire easier. Ease of installation is significant as labor costs are the greatest component of cable installation.

Use of existing wire:

The existing unshielded twisted pair voice system in a building may also be able to support data. This virtually eliminates the large cabling cost component of a LAN and the need to support two wiring infrastructures.

Ease of network cable administration:

Administration of phone systems is a familiar concept to many companies. In addition, only a single type of wiring needs to be administered.

As network hardware and software costs continue to decrease compared to the relatively constant cost of wiring, the cost of adding a LAN to an appropriately cabled building will also continue to drop.

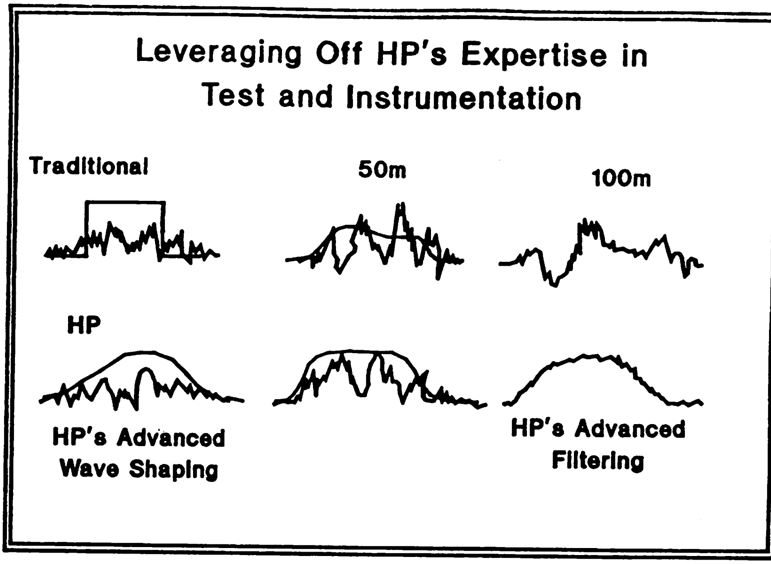


Diagram 3

Wave Shaping/Filtering

HP has addressed several technological challenges in order to provide a high-speed networking solution over unshielded twisted pair wire; these include: attenuation (weakening of the signal strength), radio frequency interference, and protection from electromagnetic susceptibility. HP has employed its technical expertise in test and instrumentation to discover techniques that allow a reliable transmission rate of 10 Mbps over unshielded twisted pair wire. Wave shaping and filtering techniques have been used to preserve the signal integrity while meeting FCC emissions requirements. Through the use of these techniques, HP can ensure an effective solution.

When you want to utilize an existing unshielded twisted pair wiring system whose condition is unknown, a wire test may be necessary. HP offers a unique service, called HP WireTest, which strengthens our StarLAN offering by providing you with an evaluation of the suitability of your existing unshielded twisted pair wiring for a StarLAN or StarLAN 10 network. The availability of HP WireTest reinforces HP's commitment to the development of complete networking solutions.

Your wiring is an investment. HP StarLAN and HP StarLAN 10 allow you to make the most of this investment by using it for LAN connectivity while not impacting the telecommunication services it currently provides.

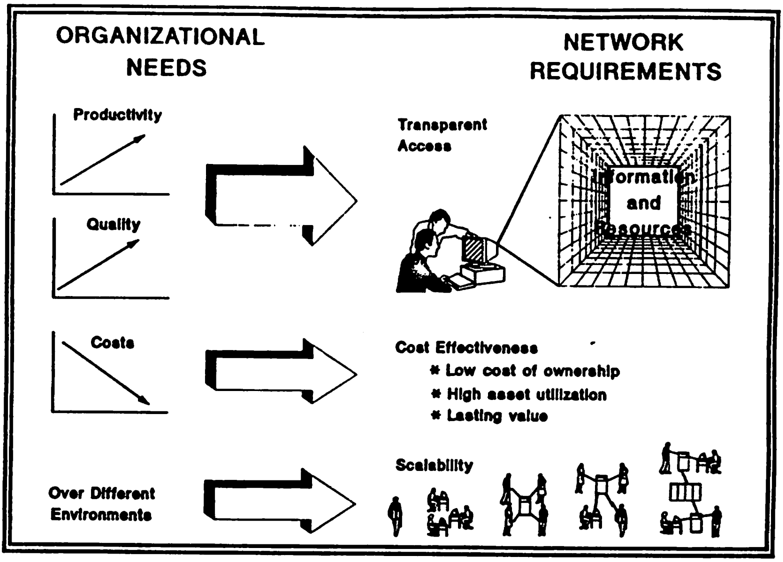


Diagram 4

Organizational Needs/Network Requirements

Meeting Organizational Needs

The HP AdvanceNet Strategy addresses organizational needs through effective networking solutions providing you with a scalable, cost effective window into the world of information and resources and ensuring a growth path for the future. StarLAN and StarLAN 10 provide the physical connection that gives users access to this world of information and resources.

HP StarLAN and HP StarLAN 10 Capabilities

When HP StarLAN or HP StarLAN 10 are used with the OfficeShare Family of Networking Software and Business System Plus, you benefit from a star topology and unshielded twisted pair wiring and have access to powerful capabilities from a friendly PC interface. PC applications such as Lotus 1-2-3 (R) and AdvanceMail are combined with HP 3000 applications such as Information Access, HP DeskManager, and Resource Sharing to provide excellent PC/Mini integration. Users have access to the resources and information they require, all from a single familiar interface, the PC.

Specific capabilities offered by this solution include:

- Transparent, programmatic PC access to information and applications on the HP 3000

- Access to transparent disc and file sharing

- Printer and plotter sharing with spooling

- Unattended backup and restore

- PC interface to a powerful electronic mail system

- Terminal emulation with file transfer from personal computers to HP 3000 Systems

- Transparent disc and file sharing and printer and plotter sharing with spooling on a personal computer server.

This combination of network and system hardware and software also benefits the MIS department by providing ease of installation, simple support, automatic updates to PC software, and centralized backup.

Advantages of HP's Solution

The use of a solution that combines a star topology and unshielded twisted pair wire with powerful applications such as HP OfficeShare and Business System Plus also provides a cost effective solution. Areas that illustrate this include:

Existing wiring may be used for networking:
The large cabling cost component of a LAN may be eliminated entirely.

Network management and troubleshooting is simplified:
With a star topology troubleshooting is more centralized, better security is provided; faults can be contained to a single node; and adds, moves and changes are greatly simplified. In addition, Business System Plus provides a utility that simplifies the addition of PCs onto a network.

Shared peripheral utilization:
People can make better use of expensive peripherals when they are easy to share.

Centralized PC software update services:

PC software management is centralized with Business System Plus. Saving considerable time. In addition, centralization of PC software updates provides version consistency throughout an organization.

Access to electronic mail:

Electronic mail provides cost savings by eliminating phone tag. The increased effectiveness of communication can improve the decision-making process.

Data backup:

We recognize that it is vital that companies protect against loss of data. Business System Plus provides automatic backup capability for personal computers ensuring protection against data loss.

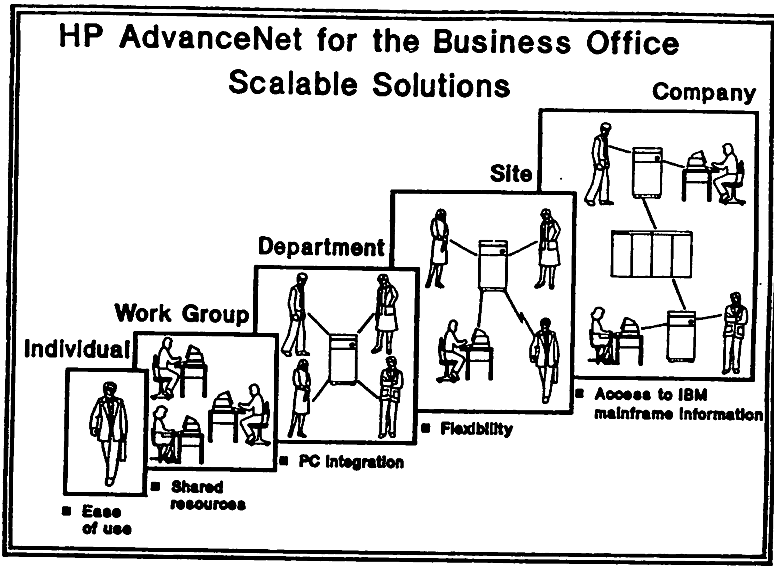


Diagram 5

Scalable Solutions

Scalability of HP Solutions

HP StarLAN and HP StarLAN 10 networks are scalable, providing growth capability as your workgroup grows and additional LAN connectivity and capabilities are required. Additional users and PC servers can be added as required. Users can begin using a PC Server and migrate to an HP 3000 server when they need increased functionality and access to additional resources and data. Users of HP StarLAN and HP StarLAN 10 can access information and peripherals residing on multiple hosts. This allows users to consolidate information which may be located on different systems, aiding effective timely decisions. Hewlett-Packard is committed to providing intuitive, transparent user interfaces to complex, powerful capabilities. As capabilities and equipment are added to the network, the impact on users is minimized, requiring minimal re-training and down time.

HP can also provide your company with comprehensive network support services. These services include analysis of your company's needs, assistance in defining a network strategy that addresses those needs, network design services, on-going training, network management and maintenance services.

Performance

In providing networking solutions, performance is an important consideration. The primary factors affecting LAN performance are:

- The network software being used

- The types of applications each user requires

- Disc performance

- The number of users requiring simultaneous access across the network.

- The CPU speed of the processors being used

- The load on each system from non-network applications

In an office environment a 1 Megabit-per-second link provides ample capacity for typical user activity. A user on a small network may perceive little or no performance difference between the 1 Mbps speed of HP StarLAN and the 10 Mbps speed of HP StarLAN 10. StarLAN 10 provides performance benefits in environments where heavy network traffic, a large number of nodes or high performance workstations exist.

Commitment to Standards

HP's commitment to standards is reflected in both StarLAN and StarLAN 10. HP StarLAN conforms to the IEEE 802.3 (Institute of Electrical and Electronic Engineers) Type1BASE5 specifications. Hewlett-Packard has been leading the standards effort to develop an IEEE standard for a 10 Mbps LAN over unshielded twisted pair wire, currently known as 10BASET. The HP implementation of this technology is now the basis for the 10BASET standard, meeting IEEE economic and technical feasibility criteria and receiving IEEE approval with respect to broad market potential, compatibility, and distinct identity.

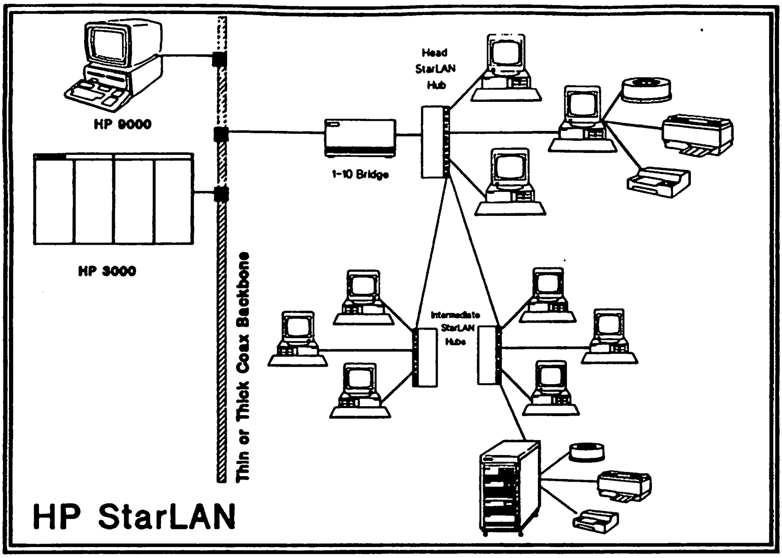


Diagram 6

HP StarLAN

HP StarLAN, StarLAN 10 and Mixed Cable Networks

HP StarLAN

HP StarLAN provides a low-cost networking solution over unshielded twisted pair wire for Business Office Environments.

The hardware components that comprise an HP StarLAN network are shown in Diagram 6. In an HP StarLAN network, each PC user workstation and PC server requires an HP StarLAN interface card and the HP OfficeShare User Services and Configuration/Diagnostics Software. Each PC server requires Server Software. HP Micro 3000 systems can attach directly to a StarLAN network with a LAN Interface Controller (LANIC). PC user workstations, Micro 3000s and PC Servers (nodes) are connected to the HP StarLAN Hub with unshielded twisted pair cable. The maximum distance between a node and an HP StarLAN Hub is 250 meters. Hewlett-Packard supports two levels of hubs with HP StarLAN. The head hub can support connections to a maximum of 11 nodes or intermediate hubs. Each intermediate hub can support connections to 11 nodes. The maximum number of nodes that can be connected to establish a standalone StarLAN network is 50.

To interconnect HP StarLAN subnetworks, which include the head hub, intermediate hubs and the nodes below the header hub, use coaxial backbone cabling and the HP StarLAN-10 Mbps 802.3 Bridge. The bridge supports communication between nodes on a StarLAN subnetwork and other systems or subnetworks connected to the backbone cable such as ThinLAN or StarLAN 10. The HP StarLAN-10 Mbps 802.3 Bridge also provides important address filtering, ignoring traffic which is not intended for the subnetwork. Nodes on a StarLAN subnetwork can transparently communicate with nodes on a ThinLAN or StarLAN 10 subnetwork.

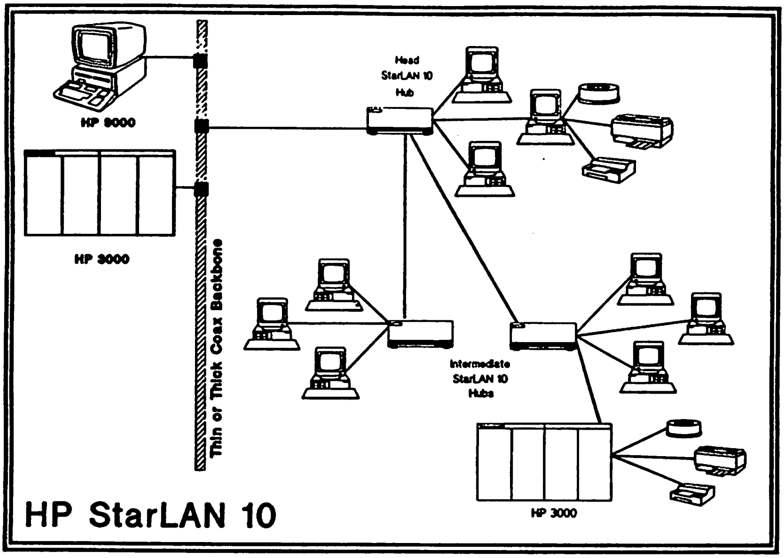


Diagram 7

StarLAN 10

HP StarLAN 10

StarLAN 10 provides a high speed twisted pair solution in Business Office, Engineering and Manufacturing environments and transparently supports a variety of networking software. The hardware components that comprise an HP StarLAN 10 network are illustrated in Diagram 7.

Each Vectra or IBM PC in a StarLAN 10 network requires an HP StarLAN 10 Interface Card. The StarLAN 10 Interface Card has one 8-pin modular plug and one 6-pin modular plug, providing for PC and phone connectivity. Other computer systems (including the HP 3000, HP 1000, and HP 9000) require a LAN interface card with an IEEE standard 15-pin AUI connector. An HP Twisted-pair MAU provides the connection from the AUI port on the LAN Interface Card to the twisted pair wire. Each node is tied to the HP StarLAN 10 Hub with unshielded twisted pair cable.

The maximum supported distance between a node and an HP StarLAN 10 Hub is 100 meters. Hewlett-Packard supports three levels of Hubs with StarLAN 10. However, we recommend limiting hubs to two levels. IEEE limitations specify a maximum of four StarLAN 10 Hubs between any two communicating nodes. (Each StarLAN 10 Hub is considered to be a full repeater.) When cascading three levels, this limitation is quickly reached. Each StarLAN 10 Hub can support twelve twisted pair and one AUI connections.

The maximum number of nodes supported by a 2-level StarLAN 10 subnetwork is 144. Multiple subnetworks can be linked via a 10 Mbps coaxial based backbone cable. This connection is made via a MAU and tap or a 10 Mbps-10 Mbps LAN Bridge may be used to connect to the backbone providing filtering capabilities for the subnetwork.

Mixed Cable Networks

HP networks are designed so that several types of network cable can be combined into one logical network. In Diagram 8, any computer can communicate with any other computer on this network. The computer can use servers, transfer files, and use terminal access. For example, a ThinLAN user could share a printer on the StarLAN 10 PC server or store files on the HP StarLAN PC server disc. Mixed cable networks may be useful when you wish to:

- Connect StarLAN, StarLAN 10 and ThinLAN subnets together directly.

- Provide access to an HP 3000, 9000 or 1000 connected to a backbone cable to users on a subnetwork.

- Connect StarLAN subnets together for more than 50 users or 1000 meters.

- Connect two-level StarLAN 10 subnets together for more than 144 nodes or 400 meters.

- Connect ThinLAN subnets together for more than 116 nodes or 740 meters.

- Connect remote PCs to an HP 3000 server via SERIAL Network along with local PCs on StarLAN, StarLAN 10 or ThinLAN.

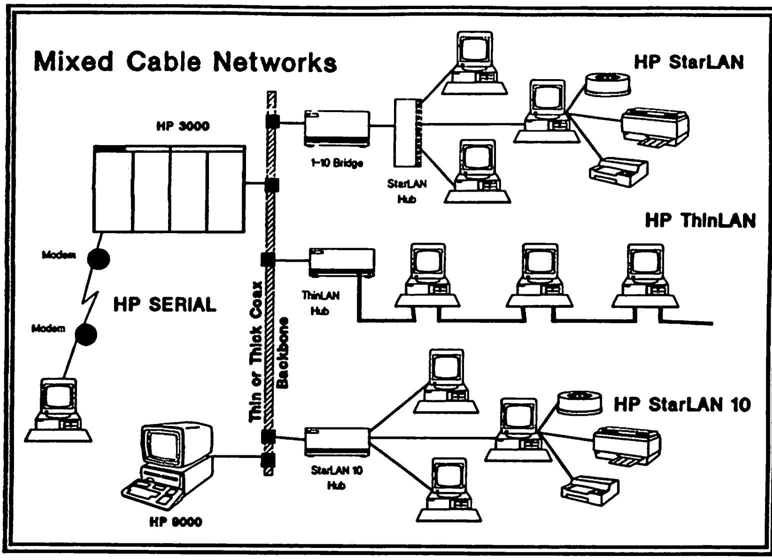


Diagram 8

Mixed Cable Networks

As shown in Diagram 8, a StarLAN subnetwork can be attached directly to ThinLAN backbone cable with a StarLAN-10 Mbps 802.3 Bridge with a ThinMAU. This connection can also be made using a ThickLAN backbone.

A StarLAN 10 subnetwork can be attached directly to ThinLAN or ThickLAN backbone cable. A MAU and tap are run from the backbone cable to the AUI port on the StarLAN 10 Hub.

A ThinLAN subnetwork can be connected to a ThickLAN backbone via an HP ThinLAN Hub. A MAU and tap are run from the ThickLAN backbone cable to the AUI port on the HP ThinLAN Hub.

Summary

HP StarLAN and HP StarLAN 10 provide several advantages through the use of a Star topology and unshielded twisted pair wire. When combined with OfficeShare software and Business Systems Plus applications a powerful, cost effective, scalable solution that effectively meets the needs of users in a Business Office environment is provided.

[1]M. Kerstettrer, "A Product Model for Departmental Computing: How do the Major Vendors Stack Up?", Small Computer Systems (Gartner Group, Inc., Feb. 25, 1987, p. 34.

[2]"Structured Distribution Systems", Local Area Communication (Gartner Group, Inc., January 30, 1987, Key Issues K-CBL-319.1

DISC INTERFACES FOR MPE-XL SYSTEMS

Gary Vogelsberg
Hewlett-Packard Company
Disc Memory Division
P.O. Box 39
Boise, Idaho 83707

INTRODUCTION

With the introduction of HP-FL, Hewlett-Packard's new fiber-optic link, there are now two disc interface alternatives for HP 3000 MPE-XL systems (they are also supported on Series 800 HP-UX systems). This paper is intended to give the HP 3000 computer user greater insight into those disc interface alternatives and how they work. It will also provide some guidelines for deciding which interface is appropriate for the user's system.

WHAT IS A DISC INTERFACE?

The disc drive is a storage device, used to store data so it can be accessed by the CPU. Internal to the CPU, data is stored in memory. There are various hierarchies of memory within the CPU, allowing the CPU to be designed for optimum price/performance. Because CPU memory is expensive and volatile, disc drives are used to complement CPU memory. They provide a place where large amounts of data can be stored in a non-volatile form, and be easily accessible to the system.

The disc interface is the connection between the disc drive and the CPU. The interface provides a means of moving data between disc mechanisms and memory within the CPU. Examples of disc interfaces on HP systems include the parallel differential interface on the 7906/20/25 disc drives; HP-IB, which is used on many current products; and HP-FL, which was introduced earlier this year and will be shipping soon on MPE-XL systems. The components of a typical disc interface are pictured below.

```
 Disc Driver            Protocol               Firmware
 -----------                                 ------------
|    CPU    |             Cable             |    Disc    |
| Interface |-------------------------------| Controller |
|           |                               |            |
 -----------                                 ------------
```

Driver/Interface Card

Within the CPU, hardware and software work together to manage disc I/O's. The disc driver is the software portion, while the interface card is the hardware portion. The interface card plugs into the internal CPU bus. The purpose of the interface card is to receive I/O requests issued across the CPU bus by the driver and to communicate those requests to the disc controllers. The interface card then manages the communications necessary to complete the I/O request. This includes managing the transmission or reception of the data as packets across the cable.

The intelligence of the interface card varies. Microprocessors can be placed on the interface, effectively downloading some of the functions off the main CPU. This can result in less CPU overhead and improved performance.

Cable/Protocol

The cable is the physical connection between the interface card and the disc controller. The cable traditionally consists of multiple wires, and is used to send commands and data between the interface card and the disc controller.

Commands are sent across the cable using a protocol. The protocol can be defined as the language used by the interface card and the controller as they communicate across the cable.

Controller

The disc controller's function is to provide the intelligence required to execute commands issued across the interface. The main functions of the disc controller are to decode commands, execute the commands, manage error recovery when necessary, manage the transmission of data across the cable, and report the execution status of commands back to the host. The controller can also be designed to provide diagnostics to aid in troubleshooting a disc drive.

HOW THE COMPONENTS WORK TOGETHER - THE TYPICAL DISC I/O

Now that the pieces of a disc interface have been defined, let's take a look at the execution of a typical disc input/ouput (I/O) request. The I/O request is initiated by the CPU when it discovers that it needs data that does not reside in memory, or that some data needs to be posted to disc. The CPU decides what data it needs to transmit/receive, and then issues a command across the CPU bus. This portion of the I/O is referred to as CPU overhead.

Upon receipt of the I/O request, the interface card looks for idle time on the bus and sends the command down to the disc controller. The disc controller then acknowledges receipt of the I/O request. The time it takes the interface to receive and then issue these commands is referred to as interface overhead.

After the controller receives the command, it decodes the command and issues a command to the disc to seek to the correct location. The time required to do this is referred to as controller overhead.

The disc mechanism must then execute the requested command. It does this by means of a mechanical movement of the actuator, placing the heads in the disc drive over the correct track. Once the heads are over the correct track, additional time is required for the disc surface to rotate so that the correct data is under the head. The time required to move the heads to the correct position is referred to as seek time, and the time spent waiting for the correct data to rotate under the head is called latency.

Once all of this is complete, data can be transferred. Data is sent to or from the disc mechanism through the interface, cable, and controller. Once the data transmission is complete, the controller sends a message that the trasaction has completed properly. The time spent transmitting data to/from disc is known as transfer time.

In a normal disc transaction on current HP systems, the majority of the time is spent in seek and latency. CPU overhead, interface overhead, and controller overhead are generally small components of the I/O execution time. Transfer time is variable, and is dependent on the amount of data being transferred.

HP-IB COMPONENTS

HP-IB disc drives have now been available on HP systems for over eight years. This interface, offering one of the first intelligent controllers, is based on the IEEE 488 electrical specification and the CS-80 protocol. This section will give a short overview of the hardware components of the HP-IB interface.

There are three basic hardware components of an HP-IB connection: the HP-IB interface card, the HP-IB cable, and the HP-IB disc controller. The HP-IB interface card plugs into the CPU bus. The HP-IB interface card provides a means of attaching multiple types of devices to the CPU. Supported peripherals include disc drives, tape drives, printers, and plotters. There is an HP-IB connector provided with the interface card, allowing the connecting of the HP-IB cable.

The HP-IB cable is a multi-wire copper cable which plugs into the HP-IB connector on the interface card. HP-IB provides 8 device addresses, so up to 8 peripherals can be daisy chained together off one HP-IB interface card. The open back of the HP-IB connector provides for easy daisy-chaining of cables.

The HP-IB disc controller resides in the disc drive. The HP-IB cable plugs into the controller. As I/O requests are received by the addressed controller across the HP-IB cable, the controller accepts and decodes the requests, and gives commands to the disc mechanism to execute the requests.

The CS-80 command set is the protocol used to communicate with discs across the interface. All hardware components of the disc interface are designed to communicate using this protocol.

HP-FL COMPONENTS

HP-FL, Hewlett-Packard's fiber-optic-link, was introduced earlier this year, and will be shipping on MPE-XL systems in the near future. Since the new interface is only recently introduced, this section will give a more in-depth view of the components of the HP-FL interface, outlining how they work together.

Cabling

There are two methods of cabling incorporated into the new HP-FL architecture. A fiber-optic cable is used to connect the CPU to a group of disc drives, while a multi-wire PBus cable is used to daisy chain a group of disc drives together.

The fiber-optic cable is a duplex cable of glass fiber; there are actually two fiber-optic strands in the cable. One strand is used to transmit data in the direction of the disc drives, while the other is used to send data back to the CPU. The fiber-optic cable has a burst transfer rate of 5 megabytes per second, and is supported in lengths of up to 500 metres.

The PBus cable is a 64-pin copper cable. It is used to daisy-chain up to eight drives to the fiber-optic cable. The PBus cable is designed to operate at a burst transfer rate of 5 megabytes per second, the same transfer rate achieved across the fiber-optic cable.

The HP-FL Interface Card

The HP-FL interface is a CPU resident card, providing an interface between peripherals and the CIO backplane on HP PA systems. Each HP-FL interface card has an improved CIO backplane interface circuit, protocol controller, high-speed parallel/serial/parallel converter and encoder/decoder, and high-performance microprocessor. A pair of fiber-optic connectors is also included with each HP-FL interface card.

The fiber-optic connectors provide a means of attaching the fiber-optic cable to the interface card. As previously noted, the fiber-optic cable consists of two fiber-optic strands. One of the connectors, equipped with a fiber-optic transmitter, is used to transmit data from the interface card to the group of disc drives. The other connector, equipped with a fiber-optic receiver, is used to receive data transmitted from the group of disc drives.

The function of the electronics is to convert electronic signals into a format allowing them to be communicated across the fiber-optic cable. When data is to be sent from the CPU to the disc drive, the interface card accepts the data issued in parallel across the CIO backplane and converts it into serial signals with the proper protocol. An LED in the transmitter then sends the data, in the form of light pulses, across the fiber-optic cable. The reverse is done when signals are sent back to the CPU from the disc drive; the interface card accepts the light pulses transmitted across the fiber-optic cable, converts them into parallel signals, and forwards them on to the CPU across the CIO backplane.

The high-performance microprocessor on the board allows the downloading of I/O processing from the CPU, reducing the amount of CPU time spent executing I/O's. The microprocessor also provides the interface card with the "horsepower" required to meet the critical timing required by the interface's high transfer rate. It should be noted that a large amount of custom VLSI work (for which HP is recognized as an industry leader) was required to provide the performance and functionality of the new interface at a reasonable cost.

The HP-FL Controller

The HP-FL controller is disc resident, replacing the HP-IB controller in 7936 and 7937 disc drives. The function of the disc controller is to manage communications between the disc drives and the CPU.

Like the interface card, each HP-FL controller is equipped with two fiber-optic connectors, one for sending data and one for receiving data. The fiber-optic strand connected to the transmitter on the interface card must be connected to the receiver on the controller card, and the fiber-optic strand connected to the receiver on the interface card must be connected to the transmitter on the controller card. LEDs on the controller indicate if the fiber-optic cable is not properly connected.

The HP-FL controller also has two connectors for PBus cables. Using these connectors, up to eight disc drives can be daisy-chained together. The end drives in a chain must have PBus terminators on their unused connectors.

With two cabling methods incorporated into the HP-FL architecture, the HP-FL controller becomes the point of management of communications moving from the fiber-optic cable to the PBus cable. As serial light pulses are received across the fiber-optic cable, the controller converts them into a parallel signal that can be sent across the PBus cable to the appropriate disc drive. Signals sent across the PBus cable destined for the CPU are converted from parallel to serial so that they can be transmitted as light across the fiber-optic cable. The interface card and controller share some common electronics, since the same functions of encoding and decoding signals and converting them back and forth from parallel to serial must be executed at both ends of the fiber-optic connection.

Once the command is received at the appropriate disc drive, the controller carries out the traditional functions of decoding the commands, executing the commands, executing error correction, and reporting back to the host.

The HP-FL interface, with its 5 megabytes per second burst transfer rate, has a much higher transfer rate than the 7937 disc drive. To use the channel efficiently, the controller has been designed to allow interleaved transfers of data from multiple disc drives. This is accomplished through buffering in the controller and active management of the PBus connection. As data is read off disc, it is stored in the controller buffer. When a critical mass of data is stored, the controller arbitrates for ownership of the interface, sends its data at the full 5 megabytes per second transfer rate of the interface, and disconnects. The controller will connect/disconnect several times when executing a large transfer. This intelligent use of the interface allows it to be used efficiently by multiple drives.

Since all components have very high burst transfer rates, the components of the controller must be designed to meet critical timing requirements. The controller was designed with precision and performance in mind, using custom VLSI extensively.

PERFORMANCE IMPACT OF HP-FL

The following shows the time to process a typical MPE-XL I/O with a 7937H and 7937FL disc drive once it gets to the disc controller. The HP-FL disc controller overhead is slightly larger than that of the HP-IB controller. It should be noted that a feature called "command queueing" allows the pre-processing of controller overhead, effectively masking controller overhead on busy disc drives. Since the 7937 disc mechanism is common to both products, the seek and latency times are the same.

The typical transfer is assumed to be 8 kilobytes on an MPE-XL system, as compared to 1 kilobyte on an MPE system. Although the HP-FL interface has a roughly five times higher transfer rate, transfer time is reduced by less than half. Since the interface rate is now faster than the disc transfer rate; transfer time for an I/O is determined by the disc transfer rate (1.89 megabytes per second across one track).

	7937H	7937FL
Controller overhead	1.0 ms	1.5 ms
Seek time	20.5 ms	20.5 ms
Latency	8.3 ms	8.3 ms
Transfer time	8.0 ms	4.2 ms
	-------	-------
Total access time	37.8 ms	34.5 ms
I/O index	26.5	29.0

The comparison shows that the typical 8 kilobyte I/O will execute in 9% less time using the HP-FL interface. At first glance, it might be expected that the HP-FL interface will have an immediate impact of at least 9% on HP PA system performance. However, there are some reasons that this is not the case.

Total access time is an important measure of disc performance. However, it is not important because a given disc drive allows the system to get back to a given user 3 or 4 milliseconds faster, but because it provides a relative measure of how many I/O's a disc drive is capable of executing.

Impact on Today's MPE-XL System

In the early design phases of precision architecture systems, conscious decisions were made to improve system level performance by reducing disc I/O requirements. The transaction look-aside buffer, memory mapped files, seek aheads, and the transaction manager were all designed to reduce the number of disc I/Os. Because of these features, MPE-XL systems will require far fewer I/Os than an MPE-V system to complete the same task.

Because of the reduced I/O requirements of MPE-XL, the HP-IB interface is able to keep up with typical MPE-XL I/O workloads. The net result is that HP-FL will have little if any impact on system level performance on today's typical MPE-XL systems.

Impact on Future MPE-XL Systems

Although 'HP-FL has little impact on the performance of today's typical MPE-XL system, there are factors that will cause the disc interface to have a performance impact in the future. Some of the factors that will cause I/O loads on the interface to increase are listed below.

HP PA systems are early in their life cycle. As new systems continue to roll out, extending the family to higher performance levels, there will be corresponding increases in I/O requirements of those systems.

Along with the increases in processing power, there will also be larger mass storage configurations on those larger systems. As more gigabytes of data are stored on each interface, there will be corresponding increases in the number of I/O's executed across the interface.

With the initial release of MPE-XL, typical disc transfers are larger than they were on MPE-V systems. Since I/O loads are relatively light today, this has not yet become a factor in performance. However, as the number of I/O's increases with larger processors and disc configurations, the larger I/O's will put an added strain on the disc interface. Growing I/O requirements coupled with larger transfers will put pressure on interface transfer rates over time.

WHICH INTERFACE IS FOR ME??

With the introduction of HP-FL, there are now two interfaces for attaching drives to MPE-XL systems. Each of the interfaces offers advantages to the customer. Those advantages, as well as guidelines for making the right decision, are outlined below.

Benefits of HP-IB

HP-IB continues to provide an easy inexpensive method of attaching a wide variety of peripherals and instruments to HP systems. On HP systems, the same HP-IB interface can be used (within supported configuration guidelines) to attach discs, tapes, printers, and plotters to the system. Because it is inexpensive and can be used for multiple peripherals, the HP-IB interface continues to make a lot of sense for small systems.

The common interface on systems and peripherals also provides an easy migration path. Since HP-IB is supported on systems from the HP 150 to the HP 3000 Series 950 , there is a great deal of flexibility in moving peripherals upward as computing needs grow.

The transfer rate of HP-IB is not a bottleneck to system performance on today's HP PA systems in most situations. HP-IB will meet the performance requirements of systems that will not be growing significantly.

Benefits of HP-FL

Eight disc drives are supported on one HP-FL interface card. Because more disc drives are supported per HP-FL interface, HP supports larger mass storage configurations than with HP-IB. Up to 30 7937FL disc drives (17 gigabytes of mass storage) are supported on the Series 950 at first release, and supported configurations will be increased with future releases of the operating system.

A side-benefit of the larger number of discs supported per interface is that CPU I/O slot usage is reduced. The large configurations noted above can be built using a minimal number of I/O slots.

Long fiber-optic cable lengths also provides a great deal of flexibility in laying-out a data center. Some of you have probably experienced the challenges of laying-out large disc configurations while staying within supported cable length constraints of HP-IB. With supported cable lengths of up to 500 metres, these problems will disappear.

HP-FL is the interface of the future for high-end mass storage for Series 900 HP 3000 systems. The improved transfer rate and flexibility provide a mass storage growth path for the future. HP-FL also provides a platform for designing future mass storage solutions. Future discs for high-end systems will be HP-FL compatible.

Because data is transmitted via light, the new fiber-optic cable also reduces environmental concerns. The fiber-optic cable is immune to electromagnetic interference. As a result, long cables can be run withougt worrying about emissions from equipment near the cabling route. The fiber-optic cable also provides electrical isolation of discs from the system, and does not emit radio frequency energy that might cause interference with other equipment.

Conclusions

Today, Hewlett-Packard is offering interfacing alternatives on MPE-XL systems. Customers anticipating the need for large configurations, longer cables, a growth path for the future, or the environmental advantages of HP-FL should purchase new disc drives with the HP-FL interface. By purchasing FL drives today at a small price premium, the larger future costs of a field upgrade are avoided. It should be remembered, however, that there will probably be no immediate impact on system-level performance.

Those customers concerned about price and not anticipating a need for the additional features of HP-FL should continue to purchase HP-IB drives. If the decision is made to migrate to HP-FL in the future, they will have an upgrade kit available to make the transition (at a price premium).

Designing Performance Tools for the Every Day System Manager

Gerry Wade
Hewlett Packard, Performance Technology Center
8010 Foothills Blvd, Roseville, CA

For years computer science has held a cloak of mystery about itself. At first it surrounded the entire field. Anyone who knew anything at all about computers could speak the jargon and baffle every day folks with seemingly meaningless acronyms. Gradually this shroud has been drawn back as more and more people were required to interact with these magic boxes. Now there remain only a few "computer wizards" who profess to understand the inner mysteries of the "System Internals". It is now time to open the cloak a bit wider and cast light on yet one more area normally understood only by the wizards.

The time has come to start designing computer system performance tools to be used by normal people instead of just by computer wizards. This article will discuss some of the requirements, techniques, and goals of such an endeavor.

Who is going to use this tool?

As you might guess from the title of this paper, I feel that there are at least two categories of performance tool users. There are the "wizards" who have spent their lives learning the obscure jargon and off-the-wall acronyms common in discussions about the inner workings of computer operating systems. They have immersed themselves into the bits and bytes to such an extent that many of them have a hard time communicating with anyone except each other. There are also a lot of people who don't know much about computers and don't really want to know any more. Finally, there is a group of people we will characterize as "Every Day System Managers" or *Ed Syms* for short. (Oh dear, another acronym !!!)

The "Every Day" in Ed Sym indicates that these folks probably don't spend a lot of time pondering the trade offs in one memory management scheme versus another. They don't have the time or the need to design operating systems; that is why they bought a computer system that came with its own operating system. The "System Manager" part indicates that they are slightly more involved in the management of the computer system than the average data entry clerk. They probably understand that batch jobs use the computer resources in a different manner than interactive sessions. They no doubt have a passing knowledge of MPE's execution priority queues (but they might be a little confused about all the details on how they work).

Ed Sym is a member of a group of front line personnel whose task it is to keep their systems running at peak performance while they deal with the day to day tasks involved in managing those systems. (Some of them actually have to deal with USERS !!!)

So why are we talking about designing performance tools for this group of Every Day System Managers ? Surely they already have their hands full working with the tools they already have . Exactly! The current crop of performance tools are a real hand full. They can provide enough data to completely overwhelm most Ed Syms. It is not so much that these tools provide useless data, or even that they do not provide enough data. The problem is that they provide data in a way that takes a performance wizard to understand.

How this came about is pretty easy to understand. Initially the performance tools were written by the wizards for their own uses in understanding and tuning the operating system. In order to expose the inner workings of the various parts of the operating system many often obscure data items were collected. Many times these items were really just local variables used by some routine in the operating system and didn't have much meaning outside the context of that routine.

During the debugging phase an attitude of "better too much data rather than too little" prevailed so these early tools collected a vast array of data items. Now when the computer systems were placed in the real world, System Managers began to ask for some tool to help them answer questions about their performance. The easiest way to provide these tools was to take what already existed, add a jazzy front end to format the data and *VOILA* performance tool. Unfortunately no one took the time to see just what their intended users would need or how they would like to see the data presented. (Even if they did, most Ed Syms did not understand performance well enough to ask for the tool they really needed).

When the initial reaction to the tools came in indicating that they were too difficult to interpret, it was all too easy for the wizards to pass it off with comments like "that's why you need us to explain it to you. It is really a complex topic after all". Now before I alienate all the wizards in the audience, I'd better move on to the next topic. In summary, the existing tools suffer from an attitude that it's OK for performance to be hard to understand and a single tool can handle everybody's needs.

Now it is time to design performance tools specifically for the Every Day System Managers to use. These tools will probably not excite traditional performance wizards but that is not their intent. The tools will be written to address the needs of the Ed Syms of the world, which brings us to our next topic...

What needs should be addressed?

There are probably as many ways to group performance tools as there are people who would want to do such a thing. For the sake of this discussion we will break performance tasks into four requirement levels. A given performance tool might address one or more than one of these areas. Personal experience suggests that any tool which attempts to address ALL of these areas is probably doomed to do none of them well.

Quick Scan or "What's going on NOW!". This is probably the most common tool. It examines what is happening on the system in "real time" (you see what is happening as it happens). The most common use of these tools is when the system performance becomes poor, to attempt to place the blame somewhere. "The system is really sluggish. Who is using up all the CPU ?" This is a *reactive* use of a performance tool.

State of the System This is the next generation of performance tools. It attempts to classify a system as to what resource is causing slow downs or perhaps to plan for hardware upgrades. Questions like "Should I buy more memory or another disc drive?" or "Would I run out of CPU if I added another 120 users?" are common. Performance tools in this area usually deal with the system as a whole and concentrate on certain key "*bottleneck*" resources.

Application Tuning In this area of performance, attention is usually focused on a particular user application or situation as regards making it go faster. "How can I reduce the response time in my order entry application?" and "What can I do so I finish the nightly batch programs before noon the next day?" are typical questions. For a performance tool to be useful answering these types of questions, it should be able to focus on particular parts of the system instead of the system as a whole.

Trouble Shooting What do you do if you have an application that runs well all day, but for some unknown reason every day from 3:35-3:40 everybody using it stops processing. How about if you have two identical systems running the same application, only one is getting half second response times while the other is taking 10 seconds for each transaction. All traditional performance tools indicate that there are no bottlenecks during these times. Obviously you have a need for some serious help. This is the case where you call you local Performance Wizard and have them bring their magic bones, tom toms, and gibberish producing performance tools to your rescue.

Terms and Jargon

Jargon can be used to simplify discussions about complex topics. How much simpler it is to discuss the performance of a *CPU* instead of referring to a *Central Processing Unit* every time. Although simplifying communications might have been the first intent, today jargon is often used for just the opposite effect. If enough abbreviations and special purpose words are used in a discussion then the uninitiated can be totally excluded from understanding it. If you wish to be an elitist then it is easy to fall into the trap of using your own specialized language to prevent you from having to communicate with "just one of the crowd". It is also easier to disguise your own shortcomings if no one can challenge what you say since they don't know what it means.

The computer industry has been very good at inventing new words and developing a language that isolates its wizards from the common man. Computer performance tools are most often created by those most high wizards that understand the operating system internals. Many of these wizards are so involved in their own world that they have completely forgotten common English. It is small wonder that traditional performance tools measure obscure quantities with even more obscure names. (How many people do you know who REALLY UNDERSTAND what a "*Memory Manager Clock Cycle*" is? Would you assume it is good if this number goes UP and bad if it goes DOWN, or the other way around?).

The first step in designing performance tools, is to decide what things these tools should measure, and what names these "things" should have. I suggest that someone who has forgotten English is not the proper person to make design decisions of this type. Please don't misunderstand. I am not suggesting that all performance tools use terms found in a sixth grade reader, nor am I suggesting that all computer acronyms be eliminated. I think we might agree that most people who are interested in computer performance would understand what a "CPU" was (or at least could understand it if the acronym was spelled out one time). I am suggesting that instead of breaking up the activities of the CPU as:

MAM, PROCESS, ICS, DISP, CACHE, PAUSE, and IDLE

we use terms like:

SWAPPING, BUSY, OTHER, DISC CACHING, WAITING ON DISC, IDLE.

Note that renaming terms may involve regrouping them as well. In my example, ICS and DISP were merged to form OTHER since not all users will care to differentiate between *Handling Interrupts while on the Interrupt Control Stack* and *Dispatching the next process*. (I can hear the whines from the techno-nerd wizards right now. "But you MUST differentiate ICS and DISP or else you can't tell about gizormplatzing v09". Well, most every day system managers never even HEARD about gizormplatzing V09 and really don't need to know about it to take care of day to day business! If they DO need to deal with it, they will probably ask a wizard to help them and they can bring out their performance tool to measure things the way they want to see them).

What tools are already available?

The HP3000 is not totally void of performance tools today so it makes sense to see what is available, and more importantly how the existing tools are being used. Here is what we discovered:

Bundled (free) tools: MPE comes with a few features which can be used to glean performance data. Most users will use commands like *SHOWJOB* for "quick scan" or "reactive" performance monitors. (Often system slow downs can be explained simply by knowing how many people are logged on to the system. Sometimes monitoring times of poor system performance with SHOWJOB can detect a correlation between system performance and certain user logons). More sophisticated system managers can even make some discoveries using the cryptic *SHOWQ* command.

We found many users performing "state of the system" analysis based on the *REPORT* command and on the wealth of information contained in the *MPE LOG FILES.* The REPORT command has only two time scales represented (NOW, and SINCE LAST RESET) and often needs to be augmented by recording its output periodically and then calculating differences. We found many tools available to assist in analyzing the REPORT command and MPE LOG FILES. Some of these tools were public domain or part of the TELESUP remote support package used by HP. Many others were available from the Users Group Contributed Library Tape or from independent software vendors.

HP Services: Hewlett Packard offers a variety of services which can address different performance needs. *HPTREND* is provided with some system support contracts. This service supplies plots of system resource usage over time periods of one to twelve months. The use of these plots is usually for a "State of the System" function and for long range planning purposes. *HPSNAPSHOT* is a consulting service which will provide detailed analysis of a short period of time (one to two hours) on a system. This service is normally used in the "Trouble Shooting" area although it has been used for "State of the System" or even "Application Tuning" functions. *HPCAPLAN* is another consulting service. It attempts to answer "what if" questions about a system's performance in the future. It uses an analytic model which is calibrated to your system using data from an HPSNAPSHOT data collection. Once the model has been calibrated it can be used to forecast the results of changing loads or potential hardware modifications.

HP Products: Hewlett Packard also offers two performance monitoring tools as purchasable products. *OPT* is probably the most well known. The "On Line Performance Tool" provides a variety of features. We found it being used primarily in two areas: For "Quick Scan" analysis, it's GLOBAL and PROCESS screens were most often used to determine what was happening on the system during times of concern. For "State of the System" analysis, OPT was run in batch mode generating log files or periodic summary reports.

APS the "Application Program Sampler" is being used almost exclusively for its most obvious function, Tuning Applications. While some site use APS as an effective "Quick Scan" facility, it has its strongest uses in identifying local bottlenecks in a particular application program.

Non HP Products: We found quite a few performance tools in use from sources other than HP. Many independent software vendors are providing a variety of tools for measuring and reporting the performance of the MPE/VE based HP3000 systems. I would be doing a disservice to attempt to list them all here as I know I would leave some out and misrepresent others. I suggest you visit the vendor's areas in any user's group meeting and scan the many periodicals and newspapers for informations on these vendors.

A few tools are worthy of special mention. These tools are often found on a system, but few people understand where they came from. I speak about the many progeny of the *OVERLORD* program (both direct descendants and relatives by association). Include in this list such names as "SOO" (Son Of Overlord), "MOO" (Mother Of Overlord), "GOO" (Grandson Of Overlord) and even "SCOOTR" (Second Cousin Of Overlord, Twice Removed). These tools sprang up from a need for "Quick Scan" information and someone's willingness to spend some time with a *MPE Tables Manual* (The Talmud of the MPE Operating System) trying to satisfy that need. These brave souls faced programming in "Super Privileged Mode" and crashing many systems in order to discover the many hidden secrets of the operating system and lay them bare for all to see. That they succeeded is evidenced by the survival of their tools. The nice thing about most of these tools is that they provide a lot of needed information and they are usually free for the asking (and the knowing of someone with connections that you can ask). The bad thing is that the original authors can rarely take the time away from their regular job to support and distribute updates to these tools. And updates are required since, unlike the real Talmud, the MPE Tables Manual is being continually changed (or at least the tables themselves are being changed). Many times these changes will cause older versions of these tools to fail or even crash your system. (No, I'm not unduly picking on the authors of these tools. I was one of those hidden folks involved in the "Overlord Legacy" myself).

How is performance data obtained?

In order to understand the relationships between different performance tools, we must dip into a slightly technical area. What does it make sense to measure on a system and how do you go about measuring it? The intent of this paper is not to make you all performance wizards, capable of instrumenting an operating system. Rather it is to expose some of the different techniques commonly in use to measure performance of a system and the trade offs between them. Knowing this will arm you with the proper questions to ask when comparing the performance numbers from one tool with that from another. You should see that it is not only possible, but likely that two performance tools can measure the same quantity and get different results if they are using different measuring techniques. The temptation is to say they one of the tools must be wrong. Some might even argue that BOTH the tools may be in error, but few would admit that both tools may indeed be correct even though that is often the case. The thing that is missing in this discussion is a strict definition about WHAT is being measured and that will often involve HOW it is being measured so let's begin.

The most commonly used techniques for capturing performance data are **SAMPLING**, **TRACING**, and **COUNTING**. Lets take measuring CPU utilization to illustrate the three techniques. **SAMPLING** interrupts the system periodically and determines its current status. It accumulates the status according to what is being measured then resumes the system from the point of interruption. To sample CPU Utilization, the sampler would simply count the number of times the CPU was BUSY when it was interrupted, and the number of times it was IDLE. The CPU Utilization can then be calculated as the number of samples in the BUSY counter divided by the total number of samples (BUSY plus IDLE).

TRACING will insert code into the operating system to log the occurrence of key events as the system runs. To measure CPU Utilization, it would probably log the time that the CPU left and entered the IDLE state. (Actually it is more common to log the time the CPU was given to a particular process and the time that process had the CPU taken away from it since these are two well defined points in the dispatcher routines). Each log record will have a time stamp to show precisely when that event occurred. By subtracting the process start time from the process stop time you get the process busy time. Adding all process busy times found in a particular interval in the log file and then dividing by the elapsed time of that interval, you get CPU Utilization.

COUNTING involves placing measurement code into the operating system much like tracing does. The difference is that the code increments a pre defined accumulator or accumulates elapsed time into that accumulator each time an event occurs rather than logging the event itself. To measure CPU Utilization, the dispatcher code that gives the CPU to a process will save the time in a special place. When the process has the CPU taken away the measurement code will subtract this starting time from the current time to get the process busy time. This process busy time is then added to a global accumulator. CPU Utilization is obtained by sampling the accumulator, waiting a set interval then sampling it again. The difference between the two values in this accumulator is the total time the CPU was busy during the interval. Divide this value by the elapsed time and you have CPU Utilization.

How do the numbers from these different techniques compare ? As you might guess, TRACING and COUNTING can achieve very comparable numbers providing they are measuring at the same points within the operating system. Either can make a mistake if all places within the operating system where a particular event can occur are not instrumented. Counting requires a pre defined array of accumulators and can consume significant amounts of main memory. Tracing can rapidly fill a log file if the events being trace occur with any frequency. (It is not unusual for process launches to occur at over 1000 times per second under MPE/VE. If you are tracing CPU as described then you will be generating 2000 log records per second). The overhead for tracing high frequency events often prohibits using this technique in other than a laboratory environment.

Sampling is often the easiest technique to implement since it does not require instrumenting many places within the operating system. It is often thought of as the least biased approach to measurement since it should interrupt the system and sample it no matter what else is happening (actually this isn't true but it is a common misconception). Sampling has the advantage of counting since it doesn't log individual events, merely counters. It has an advantage over counting since the counters do not have to exist as a part of the operating system and different measurements can define different things to count.

The problem with sampling is that a large number of samples must be taken to insure statistical validity. This assumes that the environment being measured will remain stable over the sampling interval. Sampling once each second might produce a good picture of the average CPU utilization over the period of one day, but the number would be very untrustworthy to measure the CPU utilization over one minute. This is especially true on most HP3000 systems where interactive terminal applications use CPU in short "bursts". I mentioned before that sampling was considered "unbiased" since it interrupts at random times and samples whatever is happening at those times. Actually in many cases sampling is implemented by asking the system time base generator hardware to interrupt the system at a given rate. This causes two phenomenon. First the operating system can elect to disallow interrupts during critical processing phases. This guarantees that no sample will occur during these times. Second, the system time base generator is often used by processes to pause, wait for a given time. If sampling is occurring based on this same clock, then it increases the probably that whenever a sample occurs, the "pausing" processes are in the same state (just waking up for example).

If the performance measures on your tool are obtained by sampling, you should understand the relationship between sampling frequency and duration of measurement in order to obtain a valid measurement. There are many books on the subject of statistical validity so I won't explore the issue further at this time. Also be aware that sampling works best where the system activity is uniform during the sampling interval.

When would you use one technique over another ? You would use sampling when instrumentation does not currently exist within the code you want to measure and when the values being measured may not be defined until the measurement is taken. (APS/3000 uses a sampling technique to measure CPU activity on a user application).

You would use Tracing when ever you needed to maintain a strict time sequence of events to analyze a situation. This is especially useful if the analysis must isolate a single event from all the others. (For instance, if you wanted to see a particular terminal transaction as opposed to seeing the average response time and transaction rate for a process). Tracing is especially useful in laboratory environment to trouble shoot systems where a causal relationship must be established. (Parts of HPSNAPSHOT utilize tracing of IO events to analyze disc and terminal transfers).

Counting is useful when you can pre define the events to be measured and then instrument the operating system to measure them. It is useful for long term data collections since its counters can be sampled and logged quite slowly with no loss in accuracy. It will not miss events as sampling does, nor does it require massive log files as tracing does. Counting will mask most sequencing information and so is not as useful as tracing in trouble shooting operating systems. (OPT/3000 uses the HP3000 Measurement Interface which is predominately a counting technique).

How much data is enough?

There is an old folk story about a man who captured a leprechaun. According to legend, such a capture requires the leprechaun to reveal the location of his pot of gold. Knowing the legend, the man forced his captive to lead him to the tree in the forest under which the gold was buried. After the tree was located, the man discovered that he needed a shovel to uncover it. To mark the tree, the man removed his bright yellow scarf and tied it around the tree. Now the man knew that left alone, the leprechaun would remove the scarf so the tree couldn't be relocated. He also knew that despite their mischievous nature, leprechauns were honest to a fault. Before leaving to fetch a shovel, the man elicited a promise from his captive that he would not remove the scarf or move it to another tree. Thus assured, the man hurried home to get a shovel. When the man returned, sure enough, his scarf was still tied around the tree just like he left it. However there was also an identical scarf tied around every other tree in the forest. It is said that the man died penniless after spending the rest of his life digging under all the wrong trees.

The moral to this story is that it isn't enough to present the correct answer. You must also make sure that answer is not obscured by a lot of other details. In other words, it is as important to know what NOT to present, as it is to know WHAT to present.

I suggest the following approach to guide you in discovering what is useful information, and what is distracting data. (Note that I use the term *data* to mean some measured quantity and *information* to mean data to which a MEANING is attached). Instead of starting with a list of what data is available, start with what information is required. In order to know what information is needed, you will need to decide who will be using the information and for what purpose. You should not begin to develop a tool until you have the answers to these questions firmly in your mind.

How do you develop the tools?

The uses of a performance tool can vary. You can guess what Ed Sym would want to do with a tool, or if you can find Ed, you can ask them. In either case you will probably get an answer based on what is currently available. (Most answers are similar to "I need something just like XYZ only that does blah blah blah"). You have to be prepared for what happens if you build a tool that does exactly what is asked for. What happens if you go back to your lab and build the perfect XYZ that does blah blah blah better than anyone else? You hand it proudly to Ed Sym and they carry you around on their shoulders for solving all their problems. Right ?? (I guess we're all entitled to be naive once in our life). Actually, what happens is that Ed Sym takes your pride and joy and uses it gleefully for a day or so and then says something like "This is not bad, but if it only did dumdedumdedum it would be a whole lot better". (Talk about your rude shocks!!!).

Usually its at this point that you and Ed get into a shouting match with a lot of phrases like "But it DOES exactly what you asked for" and "why didn't you think of that BEFORE?". Actually this symptom isn't restricted to performance tools. You can probably see examples of it in many different areas. There are two forces at work here. First "things change" and the longer it takes to do something, the more it can change. Second "anything introduced to an environment will change that environment". The very fact that you handed Ed Sym a new performance tool will change what Ed needs in such a tool. For example, there is no way for Ed to know how they would really like a particular feature to work until they get a chance to use it and see what it really IS.

If this sounds like a "can't win" situation then take heart. I think you can solve it through a simple process of *successive refinement* and *prototyping*. All this jumble of words means is that you should PLAN on making the wrong tool the first time. The idea is to make the best tool you can, knowing it isn't the final solution then quickly get it into the end user's hands. Now let the user work with it a bit and then accept their ideas on how to improve the tool. Go back and implement those changes and let the user have the new release then repeat the process until you have the right tool.

Obviously the amount of time it takes to create a tool and to rework it must be minimized in order to achieve the final form in your life time. (If it takes you a year to implement changes then you will probably never catch up to the other changes in the environment. If you can turn a version around in a month, or better still in a week, then you have a good chance of creating the proper tool before it is obsolete). As you might guess, traditional vendor-customer relationships will probably not allow this type of working relationship. You will probably have to sacrifice on the "prettiness" of the documentation if it must be rewritten every week. The first releases of your tool should be only minimally implemented with major pieces delayed until the underlying framework is in place. Most customers would not purchase a product that was incomplete and subject to such rapid changes.

You will probably have to develop a special relationship with a few customers during the product development process. For their help in developing the product, the customer gets to influence its course and feature set. They also get to use the tool earlier than they would if they waited for it to be finished. On the other hand, the customer must be willing to quickly install and test new versions of the tool and provide accurate feed back on their changing uses and requirements of the tool. I strongly suggest that you use more than one prototyping test site if you want to develop a generally applicable tool. This avoids anyone pressing features or procedures which will not be acceptable to the general user population.

A real problem in this type of development is knowing when to quit. Your prototyping test sites will blossom with an entire lab responsive to their every whim (more or less) and soon you may find an ever increasing list of enhancements for your tool. You will have to decide on just what constitutes a tool that is viable in the commercial market. No doubt many of the enhancements on your list would be beneficial, but at some point the tool will meet enough of the marketplace's needs to warrant making it a product. You will also have to decide what enhancements are not appropriate to the tool you are building. In order to avoid a product that looks like a camel ("a horse designed by a committee") you will have to weigh each enhancement against a design goal for your tool. It is not that the enhancements are not a good idea, but maybe they are better implemented in a separate tool.

As you can see, it isn't necessary to carefully define every detail about a performance tool's final appearance before starting to starting to write it. It IS important to have a crystal clear understanding of WHO will use the tool, and for WHAT purpose. Once these items are decided, you can form the design goals against which all enhancement requests and delayed design decisions will be weighed.

An Example: (Setting the Goals)

Let us examine a real life example to illustrate the process we have described. We have already decided WHO will be using the performance tool (Ed Sym, the Every Day System Manager). We know HOW we will build the tool (Successive refinement through prototyping). Now all we need to decide is WHAT the tool will be used for. We take as our starting point the large installed base of HP3000 systems running the MPE/VE operating system. This system was chosen first because a large number of Ed Sym's out there are asking for a more appropriate performance tool, and because the operating system is fairly stable and well instrumented for performance data. Given this wealth of performance tools on the HP3000, why even bother creating yet another one ? Why indeed ? It seems that we were constantly being hammered to provide better performance tools which were supported by HP and filled out the performance offering. To sum it all up, people were asking for a tool they could purchase for a reasonable price that would provide more performance *information* on their systems.

As I mentioned before, the specific requests were based upon what was currently available. "We need a tool like OPT but which will log more data over long time periods and doesn't require a training class to use". "We like HPTREND but we want to produce the plots ourselves and we need more information in them". "We need to be able to 'tune' a performance monitor so it looks at our applications the way we do". " We would like HPSNAPSHOT to run all the time so we don't miss any important times, but we can't afford the disc space it requires".

Our analysis of all these inputs was that what was needed is a low overhead, continuous data collection facility coupled with a friendly and flexible user presentation module. The data collected might be used for "Quick Scan" work but since OPT was satisfying that need, it wasn't critical. "Trouble Shooting" was being successfully handled by the tools in the HPSNAPSHOT package so the new tool would not have to collect the level of detail necessary to support that function. "Application Tuning" was being handled well by APS but perhaps it needed some help in identifying which applications to analyze and when. The primary area this new tool would address is the "State of the System" functions over time periods between HPSNAPSHOT (1-2 hours) and HPTREND (1-12 months).

Our initial goals statement was to design a performance data collection and presentation tool which could be used to monitor the state of the system over periods of one hour to several months. It should be a low overhead tool which does not require large amounts of disc space or other system resources since its collection would be continuous and system wide. Much flexibility as to what was collected and how it was presented is required. The data collected should be in sufficient detail to allow Every Day System Managers to identify and analyze ninety percent of their performance problems but not so detailed as to require significant training to understand.

Example continued: (Designing to meet the goals)

Now that we had our goals statement, we could begin our work. As part of our design strategy, we kept in mind that even though we were building a specific tool, much of what we did might later be extended. (For example in choosing a data structure, we purposefully choose one that could be expanded should we have a need. This decision didn't cost much in the original design, but would have been painful if we had to retrofit it later. Also, even though the original goal is not to create a "quick scan" facility, we found that it was fairly easy to design a system which would allow one to be added if desired).

Based on our goals strategy, we selected a few prototyping test sites and began to develop a working relationship with them. We spent some time getting them to tell us what they needed in a tool and then translating it back to them to make sure we understood. We were careful at this stage to keep the discussions fairly general since we weren't ready to fix the final tool features at this time.

Armed with the inputs from our test sites, we created the first pass tool and sent them a copy. As expected, at first they all said that everything was great and it was exactly what they wanted even if a few pieces were missing and it was a bit rough around the edges. After a week or so of using the tool, enhancement requests began to roll in. We filtered and combined all the requests, decided which ones were appropriate to the goals for the tool, then decided on the best way to meet the needs. (NOTE: It is at this stage that you have to be careful to ferret out the real need which is at the root of a request. If you do exactly what is asked in all cases then you will probably not design a good tool. Don't be afraid to go back to your test sites and ask them WHY they want a particular thing and then explore alternate ways to achieve it. You should also not be afraid to refuse some enhancement requests which are not appropriate to the tool's goals. Remember, the test sites are providing inputs, but YOU are responsible for the overall design of the tool).

Example continued: (Lessons Learned)

In our example, we made many design decisions after repeated cycles between our test sites and the lab. Here are a few of the most important ones:

- A continuous data collection facility is required in order to provide long term information and allow analysis of an event after that event has passed.

- The data collection facility must be of sufficiently low overhead as to not impact the performance or usability of the system it is monitoring. While some users would tolerate as much as 10% CPU overhead, most would allow from 2% to 5% maximum.

- Data which is logged should be "pre reduced" from the raw data which is available in order to reduce the amount of disc space required. It was discovered that the processing requirements to pre reduce the data did not add significantly to the overhead of the collection facility due to the reduction in the resources needed to log that data.

- The interval at which data is sampled and logged will vary based on the nature of that data. It is often possible to choose data which requires lower logging rates and provides sufficient information for the task. (You don't really have to log every disc IO which occurred for example, if you just keep track of how many were done and by whom).

- The time retention requirements may vary depending upon the type of data being logged. It was found that *GLOBAL* data (concerning the system as a whole) was often required as much as 3-12 months after it was taken (although at the longer retention times a lower number of data points per unit time was usually satisfactory). *PROCESS* data (concerning the condition of a particular user's program at one instant in time), was rarely required for more than 1-4 weeks. At longer retention times, then volume of process specific data can become overwhelming unless it is reduced by summarizing.

- Summarizing process level information into user defined groupings at data collection time allows detailed data to be logged for those groupings which would be excessive if logged by individual processes. As an example, keeping track of disc reads and writes by logical device would be prohibitive on a process by process basis.

  If you keep read & writes (2 values) for each disc drive (maximum of 32 disc drives) for each process (current MPE/VE maximum of 628 processes can be collected) then you get 2 times 32 times 628 or 40192 values. Logging this number of samples once each minute will consume 115 Megabytes of disc each day, far too much for a continuous collector.

  When we asked our test sites what they did with this process level data, they responded that they added processes together into groups which represented certain applications. By pre summarizing this data we can reduce the volume of data significantly.

  Now you keep read & write (2 values) for each disc drive (32) for each user defined process grouping (arbitrarily set to a maximum of 20 although most users indicated that 6-10 would suffice) then you get 2 times 32 times 20 or 1280 values. Now by logging this information every five minutes instead of every minute (averages of processes don't need to be logged as often as individual processes) then you collect only three quarters of a megabyte of data each day.

- Log data only when it will be used. (Avoid logging an event if it didn't occur or if it will not be utilized during analysis). Taking our previous example, if we only log a grouping of processes whenever there was some activity in that group then we can further reduce disc (and CPU) overhead without losing any information. If a user has defined 20 groupings of programs then we typically see only 3-5 groupings active during any one five minute interval. By only logging those groupings which had non zero activities, we can reduce the disc requirements to less than 100 Kilobytes of data per day.

- Applying this principal to individual process data, we found we could go even further. (With up to 628 processes on the system, each potentially logable every minute, we could easily swamp the data collector if we logged every process every minute). We found by observing the test sites, that in the "Quick Scan" mode they were ignoring most of the processes and concentrating on just those with certain characteristics. The "*interesting*" processes were those using significant amounts of one of the system resources. If we allow the user to set thresholds, then we can only log a process if it used more than the threshold amount of a given system resource. For example we would log a process if it consumed more than 10% of the CPU during a one minute sample, or if it caused more than 10 disc IOs per second during the interval. We also discovered that we should log processes which are entering large number of terminal transactions or are experiencing poor system response times. Now, to take care of the processes which never cross any threshold, we should log every process at least two times (once when it is first created so we know when it began processing, and again when it terminates so we know the total amount of resources it consumed and the amount of work it completed). With these additions, the concept of logging a process only when it became "interesting" instead of every minute, we can preserve having a lot of information about those processes without carrying a lot of unneeded data.

Example Continued: (Presenting the data)

This pretty well defined a data collection facility which was separate from the data display facility. The data collector would be running continuously on the system (in order to not miss any important events and to be able to provide a continuum of available data). What about displaying the data ?

The first approach to displaying the data is to simply format it and print it. We found this approach to be quite limited unless you could select a specific small time interval for display. This was helpful in the trouble shooting functions, but not really what was needed for the tool we wanted to build. We did discover that many of our users were now performing the "quick scan" function not in real time as they had in the past, but on recent historical data. In fact many of them commented that it was great to be able to examine what was causing performance problems this morning, or even five minutes ago, even if you weren't logged on and running a performance tool at the time. It seems that a common scenario with tools like OPT was that the system manager would get a call that the system was running slow. They would then log on to the system and run OPT to see why. They often discovered that there was no real problem and the system was running fine at the time. When they called the user back their reaction was "Yes, it is fine now, but *you should have seen it a few minutes ago*!". Now with the continuous collector, they WERE there a few minutes ago. They could just take a look at the log file and confirm or deny the actual performance and any causes for it.

We had a good tool if the System Manager wanted to know what was happening at a particular time. What we didn't have yet was a way to tell the system manager which times were worth looking at. We rapidly discovered that the best way to examine a large volume of data of similar content was to graph it. A simple plot showing how the overall use of CPU varies throughout the day could go a long way toward identifying times of interest. If we subdivided the CPU into general classes as to what kinds of things it was being used for, then the information was much more useful. This is where we discovered what kinds of terms the Every Day System Manager would like to see. From a system manager's point of view, the CPU was used by Batch Jobs, Interactive Sessions, and System Overhead. Some system managers would like to see System Overhead have Disc Caching broken out separately since they could control this facility by enabling and disabling it.

Things really got interesting when we were able to graph one resource at the same time as another. For example, if we plotted the CPU usage, Disc IO Utilization, and Memory Manager activity all at the same time then we might make some observations about the three classic system bottlenecks (CPU, DISC, and MEMORY) and whether our system suffered from any of them throughout the day. Now if you add another plot showing terminal transaction rates and response times, we can visually correlate times of heavy terminal use and poor response times against potential bottlenecks in these areas.

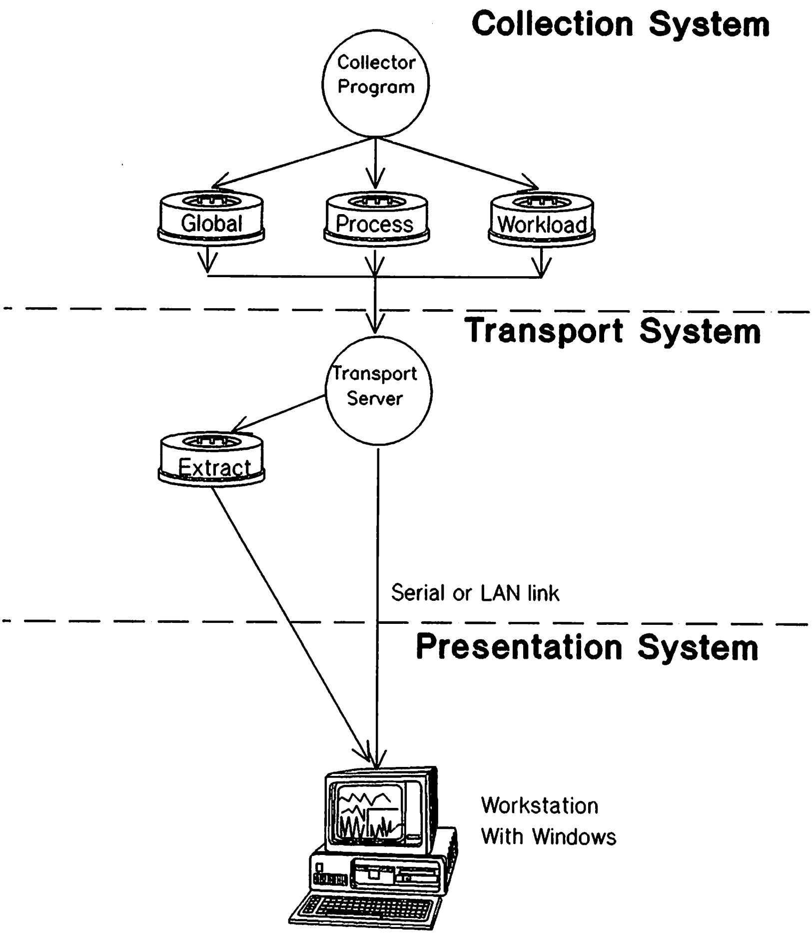
As we explored ways to present the performance data we soon found that almost every user will want to present the data in slightly different ways. Some will be concerned about spreading disc transfers across all their disc drives evenly to forestall a potential disc transfer bottleneck. They will want to show the disc activity broken down by individual device throughout the day. Others will be more interested in the response time for certain key applications and would rather graph application response times alongside overall CPU, DISC, and MEMORY usage. The possibilities are staggering considering the amount of data that is available.

In order to meet this need, we tried a "let the user's roll their own" approach. That is, let's just make all the data available and let the user define their own graph. A few of our test sites really went for this, but most said that they didn't have time to spend developing graphs and furthermore, they didn't feel good that they would graph the right things given all this freedom. Back to the drawing board. (I DID mention that the development process allowed for making a lot of false starts didn't I??)

We now have settled on using a workstation to display the data and to take advantage of the features of an industry standard windowing package to allow combining data in different ways. In order to simplify life, we have defined a set of graphs which appear to be the most generally useful from all our test site's inputs. To take care of the special cases, we allow the raw data to be subsetted from one of the standard graphs and then exported to other graphics or analysis packages (such as Graphics Gallery or a Spread Sheet). This allows the more sophisticated system manager to display or manipulate the data as they see fit, without making the original tool overly complex.

Example continued: (Final Design)

Please note that I use the word *Final* with caution since with prototype development you are allowed to make adjustments as necessary up until the product is shipped.



Data Collection:

The data collection facility will be a single process running in a batch job on the HP3000. It will gather most of its information from the MPE Measurement Interface Facility whenever possible. the Measurement Interface (MI) provides COUNTER type data for most GLOBAL and PROCESS items. The collection program will pre reduce the data before logging it. It will also form the grouping of process type data into user defined groups (called CLASSES) and log enhanced details for each class.

Some of the desired information is not available from the GLOBAL and PROCESS areas of the MI. Specifically we found that you could not obtain accurate values for Terminal Response Times, Disc Utilization Factors, or MultiProgramming Levels. The collector obtains some of this information from the MI IO Tracing facility while avoiding the overhead of tracing by analyzing the IO trace before it is logged then not logging it. Some items such as the disc utilization factor and the class MultiProgramming Level (Number of processes in a class that are waiting for the CPU) by SAMPLING at a rate of once per second. If later versions of the MI include these items as counter data items then the collector will be modified to use them in place of the tracing or sampling techniques.

The collector should be running at all times on the system being measured. It will collect and summarize its data and periodically log it to disc. Care was taken to insure that if the collector was stopped and restarted, or if the system crashed, then no significant amount of data would be lost from the log file. When restarted, the collector would continue where it left off in the original log files.

Up to three log files are used to hold three different types of data. This division of data was primarily to allow a different retention time commensurate with the usability of that data. One log file holds GLOBAL data records which report system performance on the system as a whole. Records are added to this file at the rate of one every five minutes. The second log file holds PROCESS data. A record can be added to this file once each minute for any process which is found to be "interesting". (The precise definition of "interesting" defined below). The third log file holds WORKLOAD data. One record can be added to this file every five minutes for any user defined grouping of processes (CLASS). In order to be logged, a CLASS must have had at least one process consume some CPU during the five minute sample.

The user may specify whether to log GLOBAL, PROCESS, and WORKLOAD data independently in any combination. They may also set the amount of disc space they are willing to consume to hold data of each type. The log files are "circular", that is when they are filled, old data is removed to make room for new data. In this way the user need only specify the maximum size of each file and the collector will automatically remove aged data to make room for new records. You will always have the latest data available for analysis. Since each log file can be built at a different size, different amounts of data can be maintained for each data type.

In order to reduce overhead and conserve disc space, a process will not be logged to the PROCESS log file unless it is "interesting". A process will ALWAYS be summarized into its proper CLASS and the CLASS will be logged to the WORKLOAD log file at the proper time. "Uninteresting" processes are not ignored, but their activities are summarized into classes and logged at longer intervals. A process becomes "interesting" when it meets any one (or more) of the following criteria:

- A process is seen for the first time (it is NEW).

- A process ceases to run (it DIES).

- A process uses more of a given resource than a user defined threshold value. Currently thresholds can be set for the following resources:

  CPU (default is 10% CPU over a one minute interval).

  DISC (default is 10 physical IOs per second for one minute).

  TERMINAL TRANSACTIONS (Default is 100 transactions in one minute).

  TIME TO FIRST RESPONSE (Default is 1.0 seconds from entering the data at the terminal to completing the first write back to that terminal).

  TERMINAL RESPONSE TIME (Default is 5.0 seconds from entering the data at the terminal to completing the last write prior to the next read back to that terminal).

Data Transport:

In order to present the collected data, it may be necessary to transport it to the data presentation workstation. There are several options available.

Single system connections may be made using either a Local Area Network (fastest, most expensive) or RS-232 Serial (Least expensive, slower) data comm link. Data may be analyzed directly from the three log files (even while the collector is still collecting data into them) or it may be moved to local workstation storage. The capabilities are not affected in either case, merely the access times and the amount of workstation storage required. You may also decide to use a combination of workstation (LOCAL) and HP3000 (REMOTE) data accesses. For example, if you are utilizing a low speed serial connection and you want to analyze a large amount of data, then you may download that data to the local disc storage overnight. You may still use remote access directly to the log files if you want more current data (such as data on what happened only a few minutes ago). Accessing smaller amounts of data will be relatively quick even at lower data comm speeds.

Multiple HP3000 systems may be analyzed from a single display workstation. If all the HP3000 systems are connected to the same LAN as the workstation, then the workstation can directly access the log files on any system as easily as on any other. If the workstation is connected to only one HP3000, then you can use the same data extraction tool to move selected data to the locally connected HP3000. From there the extracted files can be downloaded to workstation storage, or accessed directly from the local HP3000 disc.

As a special feature to aid those sites with multiple HP3000 systems, each log file can be automatically identified with its own user specified identification string. This identity will be displayed when the presentation package accesses each file.

Data Presentation:

A user workstation will be used to display the performance data. Industry standards are being followed when designing this display package to allow maximum flexibility and compatibility with future products. Currently the preferred workstation is an HP Vectra Personal Computer (or any fully compatible IBM PC/AT computer) configured to support MS/Windows release 2.0 or later. The PC and MS/Windows provide a rich user environment which is flexible in a standard way. It also eases the conversion to later standards such as HP's New Wave. Other workstations may be supported in later releases as these standards propagate.

Special attention is being paid to insure that the first release product will operate in conjunction with other products offered on the HP Support Workstation. (Products such as the HP LASERROM service will use the HP VECTRA PC and integral CD-ROM Disc Player).

The MS/Windows environment will make it easy for existing PC users to learn to use the data presentation software. Users which are not yet familiar with MS/Windows can follow a simple tutorial supplied with the package to learn to use it in about twenty minutes. Additional tutorial information will also be available utilizing the CD-ROM player to provide Computer Based Training. Training topics will include learning how to use the tool, Basic Performance Topics, and examples of system log files which illustrate different system situations. Using Windows, it is easy for a user to compare two or more standard data graphs by positioning each in its own window.

A Context Sensitive "Help" feature will also be included. This will allow a user to ask for help, then use a pointing device such a mouse to point to anything on the display they want to know more about (Such as the blue line on a particular plot). They will then get a "pop up" window explaining that the blue line represents CPU Busy and can range from zero to 100 percent. They can ask for more help on this or related issues or return to the graphs where they left off.

For those system managers which have a higher than average knowledge of the performance measures or who would like to perform some manipulation or alternate display techniques on the data, a data export facility will be provided. The data can be selected directly from a standard graph by simply using the pointing device to point to the start and end of the interval of interest. All relevant data which occurred in this interval will then be written to an MS DOS file in either a standard printable format, or an industry standard data transfer format such as DIF. Data heading may be included if desired to identify each data item. It is usually an easy matter to then move this data into a wide variety of PC applications such as word processing, spread sheet, statistical analysis, or graphing where additional operations may be performed on them.

Summation:

We have found traditional program development techniques to be inappropriate for the development of performance tools for every day use. In particular the design decisions for such tools are best made by a process of early prototype testing. This testing allows the users to adapt to the tools and learn what they can provide. More importantly it allows the tool to adapt to the users once they realize the potentials. This intimate feedback during the early development of a tool can result in a much more useful product and a much shorter overall time to market.

During our investigations, it became clear that performance tools designed by operating system engineers are typically not appropriate for use by end users. The fault with these tools does not always lie in their user interfaces as many believe, but rather with the choices of what data to expose. A common fault is to provide too much data thus hiding what is important. We also discovered that some attention must be paid to the terms used to describe the data. Often a simple change to the name of an item was all that was needed to clarify it. (For Example: Changing 'MAM' to 'SWAPPING OVERHEAD' does not assume that the user knows that the operating system memory management facility is called 'MAM').

A common attitude in performance logging facilities was to delay the decision of which data was important until the data reporting phase. We have found this to be a wasteful practice. By deciding what data will be required and in what formats, we were able to drastically reduce the storage overhead of the logged data. What was not obvious was that we also reduced the data collection CPU requirements. Contrary to popular beliefs, it is often less work to pre-reduce the data than it is to log all the raw data for later reduction. In this case a double savings was discovered, once during collection then again during reporting.

Substantial usability benefits are obtained by using a full time data collector. By designing a data collection facility which is low enough overhead (including the run time resource usage and in the log file storage requirements) users are not pressured to try and "catch" a performance problem while it is still happening. Since the collector is always collecting enough data for analysis, the users can relax and examine a situation after it has passed. System Managers can now afford to take a day off without having to worry about the system performance while they are gone. If any problems arise, they can be examined calmly when the System Manager returns.

The user interface features provided by a workstation such as the Vectra PC make it a lot easier for most users to gain access to their data. The use of a powerful industry standard interface such as MS/Windows (and soon New Wave) allows those users already familiar with this environment to use the performance tool with little or no training. For those users which are not already using MS/Windows or even a PC we have developed a simple tutorial. Our test site users were able to pick up the fundamentals of this environment from a short (20 minute) session following simple examples from the tutorial. Once windows fundamentals are understood, the performance tool becomes very simple to use. Each different context which the user can enter will have familiar windows features which all operate in an identical manner. Thus a user can go into the "Disc Details" context for the first time and notice that they are presented with a horizontal and a vertical "scroll bar". Even though they may not be 100% sure what each scroll bar does, then know enough to point to the bar and "click" to scroll backward and forward through the data. Additional details can always be found in the powerful "help" facility.

By separating the data collection facility from the data presentation facility several benefits were realized. First as already mentioned, we can take advantage of the powerful user interface provided on the workstation even though data must be captured on the main system. Second, multiple data collection facilities can be supported by a single data presentation facility. This means you don't have to duplicate the data presentation hardware and software for each system you are monitoring. Third, although not quite so obvious from the discussion in this paper, the workstation data presentation facility can be easily made to be independent on the type of system which collected the data. It would be possible to analyze data collected under different operating systems and architectures (say MPE/VE and MPE/XL and even HP-UX) using the same workstation. It might even be possible to examine the resource consumption of an HP-9000 running HP-UX and compare it to the same time interval for an HP-3000 running MPE/XL. To the user, it would simply be two windows on the screen, each one being controlled in the same manner.

Last but not least, we have discovered a wealth of additional uses for the long term performance data collected by the data collection facility. We can now begin to prototype different data analysis and presentation tools which use the same log files. The richness of this concentrated system performance data covering such long time intervals has surprised even our staunchest designers. Watch this space for additional details.....

Understanding IBM's

System Network Architecture

or

"Through A Glass Darkly"

A Tutorial

Robert S. Yori
Hewlett-Packard Co.
P.O. Box 152030
Irving, Texas 75015

System Network Architecture (SNA) is IBM's strategic direction with respect to data communications.

SNA is the means whereby multiple dissimiliar hardware platforms can share information. Access is provided for terminals, personal computers, and systems to communicate with each other.

The flexibility of SNA has steadily increased since its introduction in 1974. In addition to reviewing the evolution of SNA, one of the latest enhancements will be discussed - specifically peer-to-peer communications.

It will become evident from this discussion that SNA is a hierarchical communication technology. Not until the developments introduced in 1987 was true peer-to-peer communication possible in an SNA environment.

The major topics discussed here will be:

1. Evolution of SNA

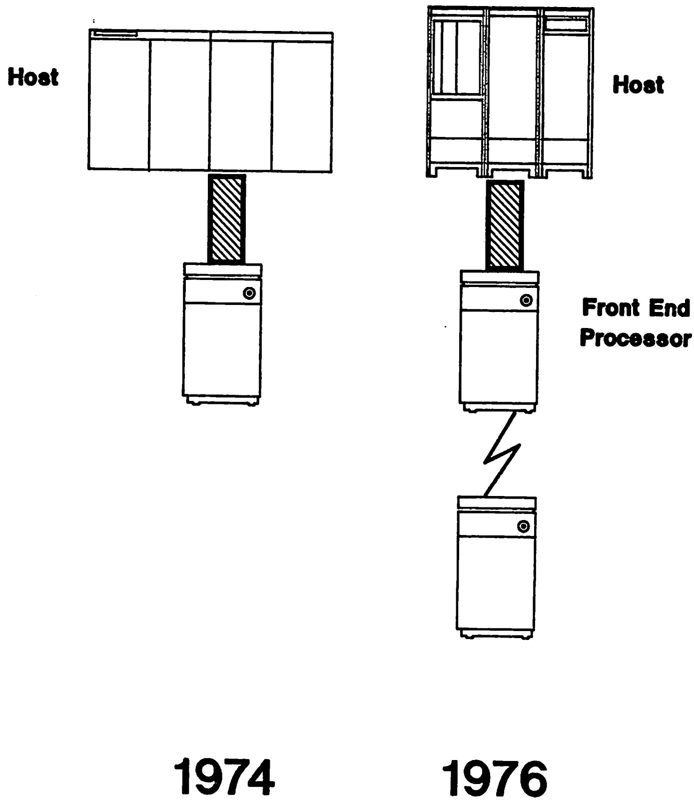
2. HP and IBM Network Comparisons

3. SNA Terminology

4. SNA and the ISO Model

5. Appendix - Creation of a Connection Between an IBM Host and an HP3000

Evolution of SNA



HEWLETT PACKARD

Evolution of SNA
1974 - 1976

In the early 1970's, IBM had a proliferation of computer systems on the market, plus a myriad of communication software packages and protocols running on these systems.

All IBM computers could not communicate with each other due to this mixture of incompatible hardware and software.

SNA was introduced in 1974 as a way of solving this problem. The objective was to provide a ommunications method whereby all IBM computers could talk to each other, and all IBM terminals could talk to any computer.

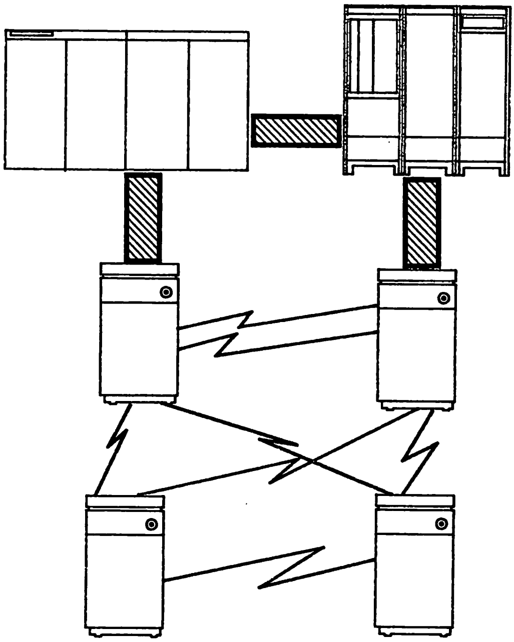
SNA's basic hardware configuration is reflected in the 1974 picture - an IBM 370 host and a front-end communications processor. Through the front-end processor (FEP), communication links are provided to remote clusters of terminals.

In 1976, the FEP's could be distributed and linked with high speed lines. Today, these are usually 56 kilobit/sec (kb) digital leased lines.

The advantage of the front end processor was that it could handle many of the basic communication tasks for large remote clusters of terminals, or remote groups of mainframes. This not only reduced the processing requirements for the main FEP, but reduced communications overhead on the leased line.

This will become more evident after the discussion of the ISO model and SNA. As a preview, the FEP's handle the communications protocols and overhead of layers 1-4 of the ISO model.

Evolution of SNA



1979/1980

HEWLETT PACKARD

Proactive Network Diagnosis

Claudia Zornow

Product Support Division
Supportability Methods Lab
100 Mayfield Avenue
Mountain View, CA 94043

I. Introduction

The term "proactivity" has come into much wider use in the last several years. For something to be "proactive" means that it is capable of dealing with problems by anticipating and preventing them, not merely by reacting to problems after they have occurred. In today's increasingly complex business and technical environment, with needs and demands for better support growing constantly, reactive support has been taken almost to its limits. Proactive support is the next step towards providing the highest possible level of service.

Support has historically been divided into the categories of hardware and software, and has been packaged and sold as such. In the last few years, as the use of networks has grown, new ways of looking at support have emerged. Network support has become an entity in its own right; it is not merely the sum of existing hardware and software support, but is the product of a new, more integrated approach.

HP's network support strategy has many facets, but this discussion will focus on only one: HP's strategy for network support tools. These tools are used by internal HP personnel in support of customer networks. The strategy used in design and development of these tools has three major components.

The first component of the strategy is remote support. HP's goal is to provide as much support on a remote basis as possible, in order to take advantage of centralized expertise of support personnel and to eliminate travel time to a customer site. The second component of the strategy is a focus on configuration. Configuration of all network elements, whether hardware or software, is a frequent source of problems with a network. Focusing on simplifying, checking and correcting configuration is an efficient way to solve a large number of network problems. Finally, the third component of HP's network support tools strategy is an emphasis on proactivity. As with hardware and software, proactivity and proactive tools for network support are the next step in providing customers with the higher levels of overall support which are required today.

This paper will discuss how HP is putting these objectives into action by developing a tool for proactive network diagnosis. It will describe in more detail why proactive network diagnosis is needed, how it operates, and what information it provides.

II. Purpose

The need for proactive network diagnosis grows out of the nature of networks. Networks are intrinsically more complex than either software or hardware, since they contain elements of both. Their many components require very specific configuration, and the configurations for the different components are highly interdependent. In addition, the networks themselves must be configured in terms of the routes, addresses, and capacities which characterize them.

The trend in networks, as in computer systems themselves, has been a steady increase in size, speed, and complexity. This trend is even more marked in networks, as the ways in which people use computing have changed. Users are not only performing more of the same activities faster, but also performing different activities. The shift towards distributed computing operations and towards the use of interconnected personal computers and workstations represents a change in what is done as well as how fast it is done. This increasing use of distributed computing creates a corresponding increase in the need for network support. Network operations in general have become a more critical part of users' businesses, and this trend will continue. Networks are also incorporating more multivendor devices, as open standards become increasingly accepted. A better way to support these ever more critical, ever more complex networks is needed.

Proactive network diagnosis is such a way. In general, "proactive network diagnosis" means monitoring events on a customer's network and identifying problems before they occur. Specifically, proactivity is provided by programmatically reading the network log files generated by the various levels of networking software and hardware, and analyzing the events and event rates recorded in the logs. The customer's system, with its network log files, serves as a "window" into the network. When the events and event rates found in the log files are analyzed, thresholds for acceptable event rates are used to determine when an exception requiring attention has occurred. Thresholds are set at a level which allows exceptions to be detected before a network goes down or becomes noticeably impaired. When an exception which can be resolved through customer action is detected, it is reported to the customer. If an exception requiring HP support personnel action is found, information about it is electronically transmitted to HP's Response Center, where the appropriate personnel can take action.

The need for this type of support is apparent. The goal of all support is to maximize the product's - in this case, the network's - usability and usefulness. This includes uptime, throughput, and reliability, among other factors. To provide this, problems need to be detected and resolved before they negatively affect customers' normal operations.

III. Operation

In order to understand how proactive network diagnosis operates and what it can provide, an understanding of the framework within which it fits is required. HP's offering in the area of proactive network diagnosis, known informally as Network Predictive, builds on the existing structure of the Predictive Support product. Predictive Support can be thought of as proactive system diagnosis. It has been running on customer HP3000 systems for several years, providing proactive support for disc drives, tape drives, and memory. Network Predictive adds network software and hardware support to the HP3000.

Products supported through Predictive Support or Network Predictive must have a high degree of internal error detection and reporting. For each product selected, product experts from the Response Centers and from manufacturing divisions model the degradation and failure modes of the product. The results are reduced to a set of rules which are incorporated into Predictive Support or Network Predictive. Each rule describes an event and specifies the frequency at which its occurrence is considered to indicate the existence of a problem; this frequency of occurrence is called a threshold. For example, a rule might be written for the network transport subsystem which describes the event "store and forward packet discarded" with the threshold "10 in a day". This would mean that, if ten or more store and forward packets are discarded in a day, Network Predictive would notify the Response Center that a potential problem has occurred. The definition of the rules and thresholds is a dynamic process involving feedback and adjustment, so the Response Center experts must monitor the effectiveness of the rules established for each product.

After Network Predictive is distributed to the customer system, analysis begins. The Predictive job is normally run nightly, during off-peak processing hours. Network Predictive executes in four basic phases. First, log files are scanned and error data is collected. Second, the error data is expressed in a generic format. Third, trend detection is performed, and finally, if necessary, the appropriate actions are taken to solve any problems. This may include automatic generation of a message to the console operator indicating what actions the customer may take to resolve the problem, or the generation of a call to the Response Center. Overall processing is controlled by the Predictive monitor process. Figure 1 shows a graphic representation of the Predictive (including Network Predictive) architecture.

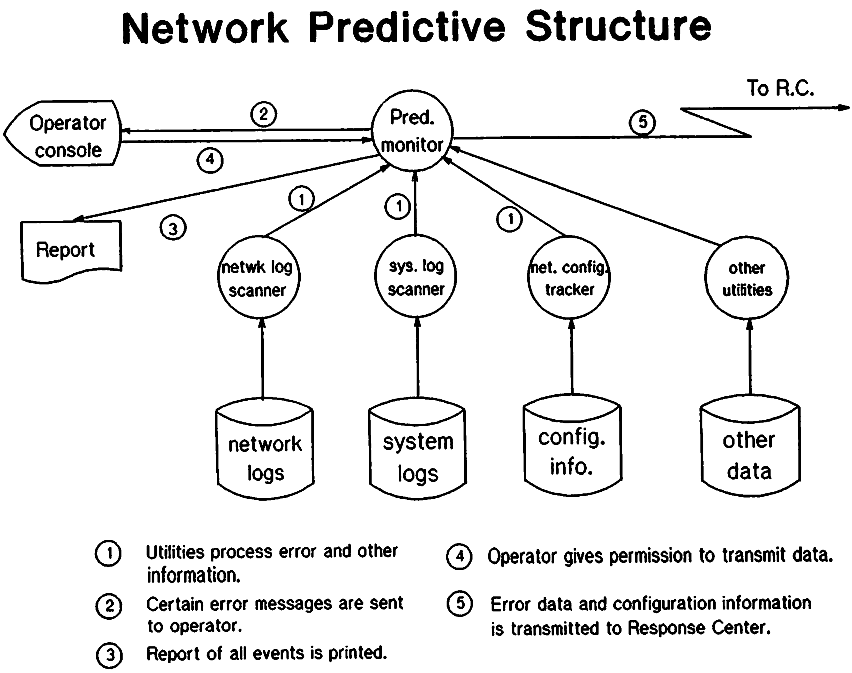


Figure 1: Structure of Network Predictive

Network Predictive uses special utility programs to collect error data. Each utility is launched as a child process of the monitor and retrieves the error data for a specific class of products, using built-in information about the format in which that class of products performs error logging. In the second phase of processing, the utilities translate the many different error data formats into a common message format, so that the Predictive monitor can use the same trend detection algorithms to determine whether a failure is imminent on the specific system component, regardless of which type of component generated the error. The messages contain information identifying the product and the type of error involved.

When the messages are received by the Predictive monitor process for third- phase processing, the error data is passed through a trend-detection algorithm. If the results indicate that an undesirable trend has been established, the appropriate action is triggered.

For trend detection, the frequency of occurrences of significant events must be established. This is done by tracking the number of occurrences of an event and weighting it by a factor such as time over which the number of events occurred, or number of related normal events (e.g. number of packets sent). In addition, since simple continuous tracking of abnormal event occurrences and weighting factor would lead to a dilution of the statistics, the accumulation of the weighting factor must be limited to create a sample across which the frequency of occurrence can be analyzed. When the trend detection algorithm detects that the number of occurrences of a given event, divided by its weighting factor and taken across the designated sample size, exceeds the threshold defined for that event, an action is triggered.

The fourth and final phase of Network Predictive processing is taking the specified action. Depending on the rule for the given event, the Predictive monitor will either send a message to the console to inform the operator about the problem, or it will use a communication link to transfer the information directly to the Response Center for investigation by support personnel. This information is then supplied to the customer in the form of a printed report.

The data communication link between the customer HP3000 system and the Response Center uses a remote support modem installed on the customer system. At the end of the nightly Predictive run, Network Predictive uses the modem link to transfer any event information gathered during that run to the Response Center. This transmission may be accomplished completely automatically, or it may be configured to require operator intervention for security reasons.

When the data reaches the Response Center, an event generation process loads it into a Response Center database for later examination by Response Center personnel. The process also generates actual Customer Service Orders within the Response Center's call tracking system, insuring that each call will receive prompt and thorough attention, including remote handling and field referral if necessary.

Network Predictive provides the capability for proactive network diagnosis. The question then becomes one of content: what information is available for predictive analysis?

IV. Network Information

Network Predictive adds a log file scanning utility to those already incorporated in Predictive Support. This new utility operates on the log files generated by HP's Network Services (NS) software. New rulesets for analyzing the data contained in the network log file have been developed. Rulesets have also been added to the set of rulesets currently used in analysis of MPE system log files; the new rulesets contain rules for analyzing the operation of INP (Intelligent Network Processor) devices. Other differences between Network Predictive and the current Predictive Support products include the increased use of rules which use a number of normal event occurrences (units) instead of time as a weighting factor, and Network Predictive's far greater number of events which generate messages to the console operator rather than to the Response Center.

Although HP's NS software was not designed to be in accord with the OSI (Open Systems Interconnect) model for standardization of networking software, the entities which comprise it can be grouped to approximate the 7-layer OSI model. Figure 2 shows the NS entities and how they fit into the OSI model. The rulesets developed for the log files generated by the NS software were chosen to operate on entities at the network, transport, and session layers. The software at these layers logs information with the most appropriate amount of detail and usefulness. Entities currently covered by rulesets include TCP SIP, TCP PM, IP, and NetIPC.

Rules by OSI levels

NetIPC - Network Interprocess Communication

TCP SIP - Transmission Control Protocol, Socket Interface Protocol

TCP PM - Transmission Control Protocol, Protocol Module

IP - Internet Protocol

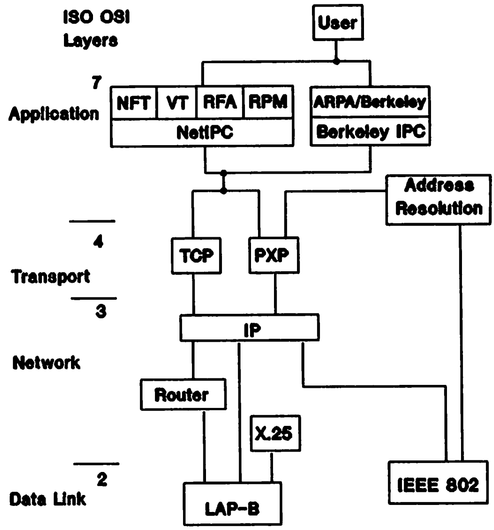


Figure 2: NS Entities in the OSI Model

The rules for the NS software logs and for the INP device information logged in the MPE system logs were developed using the technique known as "knowledge engineering". Experts in the various subject areas for which rules were being developed were identified. Development engineers working on the Network Predictive product then spent time with these experts, extracting knowledge from them and translating that knowledge into rulesets suitable for use by the Network Predictive event data collection and analysis processes. Most of the initial development of the rules was done by working with lab personnel, especially those who had been involved with the initial development of the product/entity. The rules were reviewed and the thresholds established or refined by field personnel, particularly those from the Response Center as well as the Network Marketing Center and the Customer Escalation Centers. This approach combined a thorough knowledge of product internals with practical experience of how often events are likely to occur and how serious they are. The field personnel also helped to develop the troubleshooting guidelines defined for the different events.

In their final form, the rules for interpretation of network events can be divided into five main categories: line and transmission errors, configuration errors, corrupt or defective software, corrupt data files, and hardware errors. The category of line and transmission errors includes such events as packet retransmissions and checksum errors, and usually indicates modem or line problems. Configuration errors fall into several subcategories, including resource allocation problems, incorrect address configuration, and inadequate system table allocation. These errors can normally be resolved by the customer; documentation accompanying the console messages generated by Network Predictive describes what steps the customer can take to solve the problem. Errors in the category of corrupt/defective software indicate that a software module has either become corrupted and must be restored, or contains a software defect which must be resolved by factory personnel. INP problems can also be caused by corrupt INP download code. Corrupt data files are usually network configuration files and must be restored from good versions. Finally, hardware errors indicate INP problems.

Not all of these categories are truly "proactive" in nature. The line and transmission errors, configuration errors, and INP hardware errors are the most likely to indicate impending problems. The other categories are included for thoroughness. While many of those errors are likely to show up to the customer before they are ever seen through reading a log file, some of them may pass by unnoticed, or with only a temporary workaround action being taken. Inclusion of these rules insures more complete coverage for the network.

Another important part of Network Predictive which will not be discussed in depth here is its implementation of network configuration tracking. Another utility process running under the Predictive monitor collects network configuration information from its various sources on the customer system, compares the information to the configuration information from the previous night's run, and sends the most up-to-date version through the datacomm link to the Response Center. Only changes are sent across the link once the initial complete configuration has been sent. When the information arrives at the Response Center, it is loaded into a database for use by Response Center personnel, allowing them quick access to configuration information while they work to resolve a proactive or reactive call. The Predictive Support product will be adding similar utilities in its next release, to track system configuration and software version information.

The usefulness of the network information provided by Network Predictive is dependent on the usefulness of the information logged by the network software and hardware elements. Designing for supportability is a key to successful proactive network diagnosis in the future.

V. Conclusion

Proactive network diagnosis is the next step in HP's network support tools strategy, which itself is part of HP's network support strategy: to maximize the use that customers can get out of their networks through increased reliability, increased performance, and reduced cost. By gathering the error information contained in network logs, processing it through the collected knowledge of experts, and taking immediate action on any impending problems that are detected, the Network Predictive implementation of the proactive network diagnosis concept helps to fulfill the goals of HP's network support strategy and to meet today's changing network support needs.

Considering a Network Management Service

Sandi Voykin
Hewlett-Packard
Customer Network Center
2000 South Park Place
Atlanta, Georgia

1987 was an exciting year for Wide Area Networks. Along with this new high level of interest, a new type of offering was made available to wide area networking customers: Network Management Services.

Choosing the right type of services enables you, the customer to:

1) decrease operational expenses
2) increase network reliability
3) enable internal movement of your staff to increase productivity in other areas

Failure in choosing the correct Network Management Service, or not choosing one at all could be very costly to your organization. Your network may 1) have more downtime than is acceptable to your company 2) have less than sound recommendations made on the future of your network and 3) be the reason behind a need to increase staffing, increasing your people costs. Making this decision could be critical to your organization.

To allow you to make a sound business decision while considering a Network Management Service, you need the answers to several questions. During the next half hour, we will cover questions in four distinct areas:

1) Operational procedures
2) Problem resolution
3) Customer (your own) participation
4) The Network Management Company

Operational Procedures:

Let's consider the actual offerings of this company. What hours are they willing to provide coverage for? If your organization is world wide, covering many different time zones, it is important that this Management Service provides 24 hour a day 7 day a week coverage. What about holiday coverage? In a world wide network, there will be many holidays that pertain to just a particular country and not to any others. Does this Management Service provide coverage 365 days a year? You may need that level of service, to handle the differences between country holidays.

Rarely is a wide area network based upon a single hardware vendor, so you must consider if this Management Service has understanding of a multi-vendor network. What type of working relationship have they implemented between themselves and other third party organizations? A worthwhile Management Service will have third party agreements in place, so the operational staff can take complete responsibility for problem resolution, without the customer having to place any calls to either party involved.

What type of monitoring equipment does the Management Service utilize? Is it considered state of the art? Do they actually manufacture this equipment, or is there an additional company involved? If it is an additional company, what type of response time can they guarantee this additional organization will provide in case of a hardware failure? You can not afford to have your network down due to a hardware problem on the monitoring equipment. While they are monitoring your network, is there the ability for you to also monitor the network? Perhaps in just a read only capability? You may have plans to take over the network operations in the future, and monitoring the network could be considered a training tool for your own staff.

Wide Area Networks are usually found in dynamic organizations, meaning network configuration changes must be done on a constant basis. How often is this Management Service willing to provide change management? Will this change impact your network uptime? Are the procedures documented, so you can review them? Also, is the change made, verified, documented and then the appropriate people notified (customer, local users, etc)? Is this all handled through the operations staff, even the mailings? To properly manage a network, it is crucial that the network map be accurate and kept up to date at all times.

How are you to judge if your network is running at it's peak performance? How will you determine an internal accounting structure for other divisions to reimburse you for the network time that they utilize? In other words:

What type of network statistical reports are available and how often will you receive them?

Do you pay extra for these reports, or are they part of the "operation package"? Review the reports that are provided before signing with any Management Service to be sure that they provide you with the type of data that you will need to run your business. If something is missing, can these reports be easily customized either by yourself, or by the Management Service? Is there a fee for this customization? Will you have an assigned consultant that will review these reports on a regular basis and will also come on-site for bi-monthly network review meetings? Is the consultant knowledgeable in X.25, so you will feel confident about taking his suggestions?

What type of data security is available? I realize that this can also be a hardware function, but it is important enough to be brought up here. Is the operational staff security conscious? Is the NCP (Network Control Processor) that is running your network, also running another customer's network? Or is there one NCP for a single customer?

Problem Resolution:

Now, what if you have a problem on your network? What is it you need to be aware of as far as problem resolution? First of all, is there an escalation path available? Are the procedures documented? Is there a written Disaster Plan and is it tested on a regular basis? If there is a network problem, you can not afford to have your network down while the operation staff is trying to decide what steps need to be taken. Are the lines backed up with a dial up mechanism, so your network can be running, while problem resolution continues? What is the documented response time, if you call in with a problem? Is it 2 hours, 8 hours, 24 hours, etc? How long can you afford to wait for a call back?

If the operations staff can not resolve the problem within a certain time frame, where is the second level of support? Are they physically located in the same building, so they can work together on the resolution, or are they miles apart and must deal with each other by telephone? Being separated can also add an additional time factor in the problem resolution. What if the Management Service is located in New York City and the network problem exists in Munich? What type of world wide support is available? Are there local people available, or will you have to deal with a language barrier? If on-site assistance is required, will it be local, no matter where in the world the problem is located?

Customer Participation

Now you will need to consider an area that is probably one of the most important for you: The area of Customer Participation. Let's talk about 1) cost and 2) involvement. What is this type of service going to cost you? How is it priced? Do you know up front what your monthly costs will be for the length of the contract, or will it depend on how much data you are sending? If you want to increase the size of your network, will you know before signing the original contract the approximate cost after the increase takes place? Or will you just be taking a chance?

What will be your daily involvement? None? If so, will you only need one or two people on your staff familiar with the network to make decisions? To be cost effective, you should be able to decrease your current operational staff or not hire one if you do not have one currently in place. For problem resolution, will you need a technical person available? Or is the operations staff capable of dealing with all aspects of the problem management? If you have a current staff, can you move them to other areas in the organization, to help increase productivity in those areas? The highest cost for managing a network resides in people cost. To minimize this cost, take a detailed look at what your participation will have to be.

The Network Management Company:

Let's move onto the Network Management Organization itself. What is the reputation of the company? Have you dealt with them before or do you know someone who has? Are they known to be service oriented? You will be paying this organization a monthly fee to manage your network for you, so you want to be sure that this organization is not brand new to the field. You want a company that you know has been around for a long time and will be around for a long time in the future. Have they provided this type of service before? If not, consider what this may mean to your network reliability.

Even their geographical location is important. If the management service organization is located in a large city, they will probably have access to a more rapid line installation procedure. Due to the number of lines they will be dealing with, it is important they have a good working relationship with the link vendor. After all, a percentage of network problems requires the link vendor to get involved. What is the expertise of the operations staff and what do you know of the overall quality of the people? Does this company instill a level of confidence in you? This is extremely important.

What do the actual operational facilities look like? Can you tour them at any time? When you walk through, do they look organized? Notice the building security. Network operation rooms should have key access entering and exiting. Is the room surrounded by a wire mesh under the floor and over the ceiling so no one could gain access even by climbing under the raised floor or ceiling tiles? Were you issued a visitor's pass, by a security guard upon entering the area? Were you escorted through the area, or allowed to walk around freely? What amount of your data can be accessed from the operations facilities?

Last of all, who are their other customers? Are you encouraged to contact them as a means of providing a reference? Are they a small company or a Fortune 100 company? If you do contact them, can you be positive that they will provide you with a totally unbiased account of the Management Service?

As you can see, choosing a Network Management Service is not an easy task, but making the right choice can be extremely beneficial to your organization. You can benefit by 1) lower costs 2) higher overall productivity and 3) higher network reliability and performance. I hope the questions I have mentioned will enable you to make a good solid business decision on what you need a Management Service to provide. I have also provided a handout of these questions in a format that you should find useful as a worksheet when the question of a Network Management Service arises in your organization.

OpenView Windows: HP's New Foundation for Network Management

Kathleen Gannon

Hewlett-Packard Company
Information Networks Division
19420 Homestead Road
Cupertino, CA 95014

With the announcement of OpenView Windows, Hewlett-Packard starts a new era in Network Management. In the old era, users had to learn many different interfaces for their network management utilities, and often had just as many monitors to watch in their control rooms. With OpenView Windows, network management utilities can be combined on one station, with a unified user interface based on Microsoft Windows. Furthermore, a Developer's Kit is available to allow third parties and in-house developers to develop their own applications under OpenView Windows.

This paper focuses on the OpenView Windows Developer's Kit. First, it describes the features provided by OpenView Windows to an application. Next, it discusses the overall architecture. Finally, it features sample code modules to demonstrate the use of the intrinsic and message interface.

OPENVIEW WINDOWS FEATURES

From a developer's point of view, what are the benefits of OpenView Windows? The most obvious is that its easy to use interface, based on an industry standard, improves the saleability of the developer's product. Additionally, OpenView Windows provides services to the developer that reduce the amount of work necessary to get a product to market. These include the OpenView Windows network map, status management, menu integration and help utility.

The network map is the heart of OpenView Windows. It consists of a collection of pictures, linked togther by subnet symbols. A picture contains symbols representing network components for a portion of a network. A subnet symbol is a special symbol that represents one of the pictures ("subnets") in the map. It performs two functions for that picture: first, if the user double-clicks on the subnet symbol with the mouse, the corresponding picture is displayed; second, the subnet symbol takes on the color of the most critical alarm in its corresponding picture, thus providing a status summary. Like all symbols, subnet symbols can be used freely throughout the map. Using OVDRAW, the end user can draw a map of his network as he views it, grouping nodes and networks into pictures according to whatever scheme makes the most sense to him.

The symbols in the network map perform two functions. By changing color to display status, they provide a quick overview of the state of the network. They are also the key to

---

Microsoft Windows ® is a registered trademark of Microsoft Corporation.
OpenView ® is a registered trademark of Hewlett-Packard Company.

the integrated, object-oriented command style of OpenView Windows. Users simply click on the symbol representing the network component they wish to work on, then select the desired function from the menus. Behind the scenes, OpenView Windows determines which menu items to enable, based on the type of symbol selected and the network management applications installed, then routes the user's command to the appropriate application to perform. In this way, users need to learn only one command to peform a function on many different types of network components.

OpenView Windows provides five levels of status: Critical, Warning, OK, Offline and Unknown. Once a network management application has determined, using its own methods, which of these states a component is in, all the application needs to do is tell OpenView Windows the object's name and state. OpenView Windows will change the color of all instances of that object in the map to the new state. In addition, if it is a change to Critical or Warning, a message will be displayed to the user to notify him of the event, and a time- and date-stamped entry will be made in the OpenView Event Log. A menu item is available for the user to view this event log.

One of the main factors contributing to OpenView Windows' ease of use is its ability to overlay the functions of several network management applications onto a single menu item. The user no longer has to remember different syntaxes to perform similar functions on different components. OpenView Windows has defined some generic functions that should be common to many components. If the application wishes to provide the functionality for one of the generic functions, it registers for the type of object it manages, then registers for the desired menu item. When the user clicks on the registered object and menu item, the command will be routed to the application for it to execute. If an application wishes to provide additional functionality, it can add its own menus and menu items.

On-line help has been shown to greatly improve the usability of a product. OpenView Windows provides the same On-line Help Utility as HP's NewWave product. Application writers simply write the textual help file for the application, and assign context numbers to each topic. A program is then used to prepare this file for use by the Help Utility.

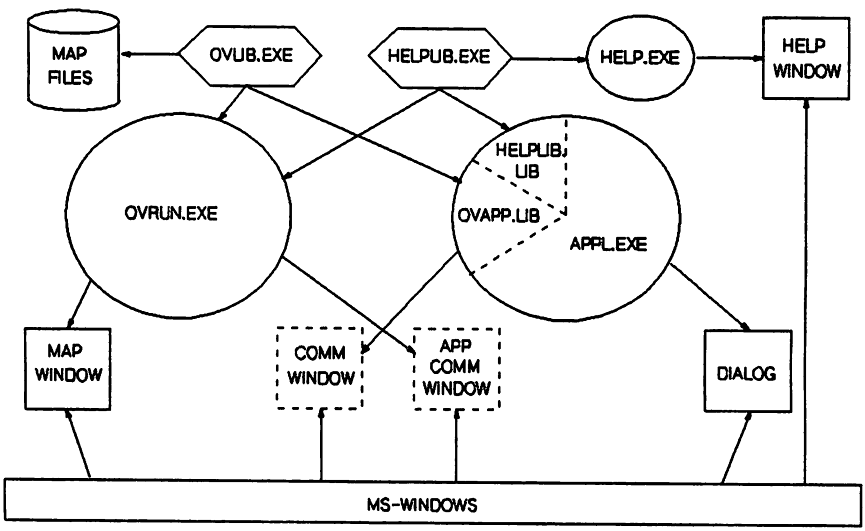
When users are running the application, they can request an index to the help topics. The application makes one procedure call and OpenView Windows handles the rest, allowing the users to browse through the topics to their heart's content. Likewise, if the application wants to show the user a particular topic, there is another call to start up the help window, automatically displaying the desired page. Once the window is displayed, all user interactions with it are managed by the Help Utility. No more effort is required on the part of the application.

OPENVIEW WINDOWS ARCHITECTURE

From the end user's point of view, there are three programs: OVRUN, OVDRAW and OVADMIN. OVRUN would typically be running continuously in a network control room, displaying the network map with the appropriate status colors. It contains functions for monitoring, diagnosing and controlling the network. OVDRAW is the utility used to draw the network map. OVADMIN is used to configure such things as passwords and defaults in the network management software itself.

When users start up one of the three programs, that program consults the initialization file, WIN.INI, and spawns each network management application installed there. Figure 1 shows the communications between one OpenView Windows program (OVRUN) and one network management application. OpenView Windows talks to the application via the Microsoft Windows message interface while the application talks to OpenView Windows via a procedural interface. This procedural interface masks the real implementation of the calls from the application. Some of the calls are implemented by the two dynamic libraries, NETWLIB and HELPLIB, while others are converted into messages. Two invisible windows act as message queues, one for the OpenView process and one for the application. The map window and the application's dialog windows are visible windows which comprise the user interface. In addition to OpenView messages, all windows will also receive messages from Microsoft Windows.

OpenView Windows Architecture



SAMPLE CODE MODULES

The following code segments were taken from a sample application called "Grape". The entire source for Grape would be too long to fit here, but a few procedures have been included to illustrate the use of the OpenView Windows programmatic interface. Procedures and messages starting with "OV" or "HELP" are part of this interface.

"Grape" performs the following functions:

1. Register for OBJ_COMPUTER symbol
2. Enable generic menu item, "Describe".

3. Add its own menu, "Measure".
4. Add its own menu item to the Measure menu, "My menu item".
5. Add a "Grapevine Index" menu item to the Help menu and enable the Help subsystem.
6. Respond to OV_GETVERSION message by returning a version string.
7. Re-start its status polling when it receives an OV_NEWMAP message.

```
/*************************************************************************
 * InitApp                                                               *
 *                                                                       *
 *    This routine is called as soon as the application starts up,       *
 *    before it drops into its WinMain loop.  Since OpenView             *
 *    Windows waits until it receives OVInitComplete before              *
 *    spawning the next application, a well-behaved application         *
 *    does as little as possible here.                                   *
 *                                                                       *
 *    The things to be done in this procedure:                           *
 *         1.  Call OVInit to create the comm window for the app.        *
 *         2.  (If desired) register for one or more symbols types.      *
 *         3.  Add menus and menu items.                                 *
 *         4.  Call OVInitComplete.                                      *
 *         5.  Initialize the Help subsystem.                            *
 *         6.  Any other application-specific initialization.            *
 *************************************************************************/
int InitApp( hInstance, hPrevInstance, lpszCmdLine, cmdShow )
HANDLE hInstance, hPrevInstance;
LPSTR lpszCmdLine;
int cmdShow;
{
        HANDLE hTask;
        int nError;
        char buf[40];
        OVPARM far * parmp = (OVPARM far *)lpszCmdLine;
        char szHelpDir[128];

        /* store the instance handle in a global variable for
           later use */
        hInst = hInstance;

        nError = OVInit( hInstance, (FARPROC)CommWndProc, lpszCmdLine,
                                (LPHWND) &hCommWnd );
        if( nError != OV_SUCCESS )
                MessagePrintf( 0, "Init %d", nError );

        /* MainAddMenu registers for an object and adds menu items */
        if ( MainAddMenu( hCommWnd ) != OV_SUCCESS)
                MessagePrintf(0, "Error adding menu.");

        nError = OVInitComplete();
        if( nError != OV_SUCCESS )
                MessagePrintf( 0, "Init-complete %d", nError );
```

```
        /* This is a Microsoft Windows call to retrieve a string
           from the initialization file, WIN.INI */
        GetProfileString( (LPSTR)"OpenView", (LPSTR)"HelpDir",
                          (LPSTR)"", (LPSTR)szHelpDir,
                          sizeof(szHelpDir) );

        /* Tell the helplib where your help text file is.  It returns
           a handle which must be used in all subsequent calls to the
           help facility */
        hHelp = HELP_Initialise( hCommWnd, hInstance,
                                 (LPSTR)szHelpDir,
                                 (LPSTR)"grape.hlp",
                                 (LPSTR)"Grapevine", FALSE );

        return TRUE;
}

/**********************************************************************
 *  MainAddMenu                                                       *
 *      Called during initialization to register to manage a         *
 *      particular type of object and to add menu items.             *
 **********************************************************************/
MainAddMenu( hWnd)
HWND hWnd;
{
    char szBuf[MED_BUF_SZ + 1];
    int iResult;
    int iMenuID;

    /* Register for the object type that the application will
       manage. */
    iResult = OVRegister( OBJ_COMPUTER );
    if (iResult != OV_SUCCESS)
        return MainError( iResult, (LPSTR)"MainAddMenu",
                (LPSTR)"OVRegister" );

    /* Monitor menu */
    /* DESCRIBE: generic menu item */
    iResult = OVMenuAddItem( 0, (LPSTR)"", OV_IDMDESCRIBE,
                        OVM_ENABLED | OVM_OBJSPEC );
    if (iResult != OV_SUCCESS)
        return MainError( iResult, (LPSTR)"MainAddMenu",
                (LPSTR)"Describe" );

    /* New menu  "Measure" */
    LoadString( hInstance, IDSGV_MEASURE, (LPSTR)szBuf, MED_BUF_SZ);
    iResult = OVMenuAdd( (LPSTR)szBuf, (LPINT)&iMenuID);
    if (iResult != OV_SUCCESS)
        return MainError( iResult, (LPSTR)"MainAddMenu",
                (LPSTR)"Measure Menu" );
```

```
    /* Add to Measure menu: "My Menu Item" */
    LoadString(hInstance, IDSGV_MYMENUITEM, (LPSTR)szBuf, MED_BUF_SZ);
    iResult = OVMenuAddItem( iMenuID, (LPSTR)szBuf,
                        GV_OBJSPEC,
                        OVM_ENABLED | OVM_OBJSPEC );
    if (iResult != OV_SUCCESS)
        return MainError( iResult, (LPSTR)"MainAddMenu",
                (LPSTR)"My menu item" );

    /* Add HELP INDEX to Help Menu, Application specific */
    LoadString( hInstance, IDSGV_HELP, (LPSTR)szBuf, MED_BUF_SZ);
    iResult = OVMenuAddItem( OV_IDMHELP, (LPSTR)szBuf, GV_HELP,
                OVM_ENABLED );
    if (iResult != OV_SUCCESS)
        return MainError( iResult, (LPSTR)"MainAddMenu",
                (LPSTR)"Help Index" );

    return OV_SUCCESS;

}

/*********************************************************************
 *  CommWndProc                                                      *
 *    This procedure handles all messages sent to the application's *
 *    communication window.  In this case, it only handles OpenView *
 *    messages, and passes on any it doesn't care about to the      *
 *    default window procedure OVDefCommWndProc.   Since it is      *
 *    an invisible window, it doesn't need to act on most           *
 *    Microsoft Windows messages, however the application           *
 *    could if it desired.                                          *
 *********************************************************************/
long FAR PASCAL CommWndProc(hWnd, message, wParam, lParam)
HWND       hWnd;
unsigned   message;
WORD       wParam;
LONG       lParam;
{
      HANDLE hMem;
      LPSTR lpMem;

      switch (message) {
        case OV_SHUTDOWN:
            /* This message tells the application that
                OpenView Windows is going away, so it needs
                to shutdown. */
            HELP_Done(hHelp);
            OVDone(hHelp);
            PostQuitMessage(0);
            break;

        case OV_COMMAND:
            /* This message results from the user choosing a menu item
```

```
            which the application had enabled. */
        MainCommand(hWnd, wParam, HIWORD(lParam), LOWORD(lParam));
        break;

    case OV_GETVERSION:
        /* This message is generated when the user brings up the
           "About" box.  The version strings of all installed
           applications are displayed in a listbox within the
           About dialog. */
        hMem = GlobalAlloc( GMEM_MOVEABLE | 0x2000, (LONG) 256 );
        lpMem = GlobalLock( hMem );
        farstrcpy( lpMem, (LPSTR)"Grapevine   X.00.00" );
        while (*lpMem++)
            ;
        *lpMem = '\0';
        GlobalUnlock( hMem );
        return (LONG) hMem;

    case OV_NEWMAP:
        /* This message tells the app that the user has loaded
           a new map, so all previous status information has
           been erased.  The application needs to restart
           whatever method its using to monitor status on the
           components it is managing. */
        MainStatusInit();
        break;

    default:
        /* Let the default window procedure handle any other
           messages that may come in */
        return OVDefCommWndProc( hWnd, message, wParam, lParam );

    } /*end switch*/

    /* A window proc should always return something */
    return(1L);
}


/*****************************************************************************
 *  MainCommand                                                              *
 *      Called when a OV_COMMAND message comes in, indicating that           *
 *      the user has selected a function from the menu.                      *
 *****************************************************************************/
int MainCommand(hWnd, wParam, wHiWord, wObjId)
HWND        hWnd;
WORD        wParam;
WORD        wHiWord;
WORD        wObjId;
{
    int iResult;
```

```
    switch (wParam) {
        case OV_IDMDESCRIBE:
            /* Display the Describe dialog box */
            NSDetailShow(hWnd);
            break;
        case GV_OBJSPEC:
            /* User selected "My menu item", so we put up a
               message box saying "My Menu Item". */
            MessagePrintf(0, "My menu item");
            break;
        case GV_HELP:
            /* User has invoked Help Index menu item */
            iResult = HELP_Index( hHelp);
            if (!iResult)
                MessagePrintf( 0, "Grapevine Help Index Error" );
            break;
        default:
            MessagePrintf( 0, "Grape_COMMAND, not recognized: %d",
                                                wParam);
        }
    return TRUE;
}

/*********************************************************************
 *  MainError                                                        *
 *      Error reporting is up to the application, but OpenView       *
 *      provides two intrinsics to help.  OVErrorMsg returns         *
 *      a text string to explain a OpenView Windows error number.    *
 *      OVLogWrite writes a message to a log file on disc which      *
 *      can be used by support personnel to trouble-shoot the        *
 *      Network Management software later.                           *
 *********************************************************************/
int MainError( nError, lpProcName, lpMsg)
int nError;
LPSTR lpProcName;
LPSTR lpMsg;
{
    char szMsg[ MED_BUF_SZ + 1];

    if ( lpMsg == (LPSTR)NULL ) {
        OVErrorMsg( nError, (LPSTR)szMsg, MED_BUF_SZ );
        lpMsg = (LPSTR)szMsg;
        }

    OVLogWrite( (LPSTR)"Grapevine", lpProcName, nError, lpMsg);
    MessageBox( 0, (LPSTR)lpMsg, (LPSTR)"Grapevine", OV_MSGERROR );

    return nError;
}
```

TITLE: SYSTEM SECURITY: Access Control in the MPE Environment

AUTHOR: Ken Jordan

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 2049

Integrating MPE XL: A True Story...

Rex Backman
Hewlett-Packard Company
Roseville, California

Introduction

The recent introduction of the Hewlett-Packard Precision Architecture family of machines presents several growth path opportunities into the next decade. MPE based environments will need to be migrated to the MPE XL based environments found on the HPPA machines. While lots of attention has been focused on the compatibility mode (CM) versus native mode (NM) arena, the successful integration of an HPPA machine requires a larger view than just the CM/NM issues found in the application systems of the local environment. A true migration needs to include the analysis and resolution in the functional areas of Operations and System Management plus the aforementioned Application area. Our shop addressed these functional issues and the results of this work was the replacement of an HP3000 Series 70 by a HP3000 Series 950 over a weekend with no true loss in user functionality!

The preparation required to perform the task of a migration from a classic machine to that of a HPPA machine is the key to the success of a migration. Proper planning, realistic time schedules, correct and adequate technical skills can be leveraged together with the end result being a transparent (to your users!) movement to a MPE XL based environment. You don't have to perform a switch over a weekend like we did, but with a proper emphasis on the aforementioned basic details, integrating an HPPA machine can be painless.

Each functional area is equal in importance. Without one, all others will be weakened in their chance for achieving a transparent migration. Attention to details in each will insure success. Knowledge of your local environment combined with the preparation mentioned above contributes to the success. Now, let us look at the major functional areas that need to be addressed. While only an overview is presented here, the items discussed here should help a site in getting a handle on their migration directions.

System Management

The changes in this area are great and operating procedures that experienced System Managers have grown accustomed to on MPE V/E are obsoleted by the tools available on MPE XL. However, the changes are augmented by positive movements in flexibility. Described below are some of the major tools that a System Manager will use, again this is a brief overview, knowledge of these skills will provide a high degree of confidence in any MPE XL installation.

SYSGEN

SYSGEN used to configure machine attributes is an interactive or batch tool with a hierarchical structure. Gone is the linear methodology of SYSDUMP. Parameters are keyword or positional with the specific areas of system configuration partitioned into menus. Finalized system configurations are "kept" as a set of logically related files in a user determined group of the SYS account. The group structure allows for multiple configurations to be in place on the machine at the same time. Activation of the selected configuration is accomplished by a simple parameter on the MPE XL boot command. Users of SYSGEN will enjoy the flexibility and functionality that does not exist in SYSDUMP. The menu driven user interface allows System Managers to easily and rapidly see their machine environment.

VOLUTIL

Disc drive characteristics are handled by the system utility VOLUTIL. This tool is analogous to the MPE V/E VINIT subsystem albeit much more powerful. Local considerations come into play here. Does the shop require non-system defined disc volumes? Or on the other hand, does the shop require all disc volumes to be MPE XL volume set members? There are advantages to both options. Selecting non-system volume domains allows for portability of data. Also, a drive can be downed on the machine without causing harm. Complete MPE XL volume set members are defined once and do not need any management of directory entries . Whatever option is chosen, VOLUTIL is the tool that formats, initializes and labels disc volume labels. Disc space storage parameters for disc files "permanent space" and virtual memory "transient space" are allocated during VOLUTIL work. Volumes can be defined before they actually reside on the machine allowing for quick and easy disc drive additions.The VOLUTIL manual has several examples of how to utilize the tool to your advantage. Understanding the logic behind Volume Management will be advantageous to new HPPA System Managers.

NMMGR

Network Management is altered by the appearance of the Distributed Terminal Controller (DTC). No more ADCCs or ATPs, all hardwired terminal connections are made via this asynchronous box. Multiple DTCs can exist on your machine and they are extremely portable. Configuration of terminal ports as well as NS XL is accomplished thru the NMMGR utility. Any network related configuration issue is handled by NMMGR. While system to system configuration is similar to that of NS/3000, the work required to configure terminal ports brings into play some new terms. Term Types are replaced by Profiles which allow NMMGR to understand the specific type of connection being configured. Modems, direct connect ports, serial printers all have different Profiles. Once site network configuration parameters are complete, a Validation function is executed from within NMMGR to allow the user to check their work for logical inconsistencies. To properly utilize NMMGR, the System Manager will need to know the hardware addresses ("paths") of the LANIC cards (XL machines use two) used for NS XL services and DTC services. Sample configuration files are provided on the MPE XL FOS/SUBSYS tape to expedite network configuration. These files contain the basic network configuration infrastructure, the user can then customize these generic files to fit their specific requirements.

These three tools are the basic tools needed by the site System Manager to get a MPE XL machine operable. Other tools complement these to provide a set of tools to allow for proper system management. Programs such as: FSCHECK for file management, DIRMIG for directory migration, DISCFREE for disc space reporting are just a few. Successful understanding of these tools is paramount if your MPE XL integration is to succeed.

Operations

Cosmetically, Operators see pretty much what Operators see currently on MPE V. Underlying this "compatibility mode" appearance are drastic changes in system Start Up/Shutdown, "Reloads", and Store/Restores. Again, much like the System Management tools, these changes are positive in nature. The flexibility, freedom, and functionality inherent in these new Operations tools overcome any initial learning curve requirements.

Starting up and shutting down a MPE XL system is a logical place to begin. The MPE XL Initial System Loader (ISL) command START is used to facilitate a system boot which saves spoolfiles. The same command modified with a NORECOVERY option is the equivalent COOLSTART command. UPDATE CONFIG replaces COLDSTART and INSTALL acts as the Reload equivalent. LDEV 1 is still "used" on START and START NORECOVERY. Remember that we can keep multiple configurations on the machine, these can be activated as needed by the GROUP parameter. The issue of CM/NM also surprisingly appears in the system boot process. The SYSSTART.PUB.SYS file needs to have its' execution blocks defined in MPE V/E terms. The operating system code does not recognize START, START NORECOVERY, etc. Instead the user can map WARMSTART, COOLSTART, etc into their SYSSTART file. Simply restoring your SYSSTART file from the classic machine will facilitate this task.

Shutdowns are more in line with Operator's MPE V/E experiences. CNTL A SHUTDOWN provides the means to properly halt the operating system. CNTL B HALT is replaced by CNTL B RS (Reset System). This command reboots the low level machine code which resets the machine's attributes such as memory. Memory Dump commands exist for the random system abort (MPE XL term for system failures). CNTL B TC preps the machine for the ISL DUMP command.

Simple machine "Reloads" do not exist as we know them in the MPE V environment. On MPE V/E machine, a simple RELOAD followed by the proper option initiates the reload process. When finished with mounting Reload tapes, the Operator has a functional MPE V/E machine. This is not the case on MPE XL. Install is the command used to initiate the Reload equivalent. The major change here is that volume labels need to be added back to the disc volumes if they are MPE XL System volume members. Also, the system directory will need to be "restored" from a NM backup tape. If a site has chosen to use non-system defined volumes then the "Reload" or Install process is much easier, however account management complexity is increased.

Stores/Restores again have a NM or CM option. Native mode stores are faster and have some of the HP TurboStore features built in such as interleaving. The NM store option is the only way to store the system directory as well. Caution should taken when selecting the NM store option as tapes created in this fashion are not portable to classic machines. CM store should be used if portability is an issue. While slower, CM store allows you to have the added capability to port files to classic machines as needed. Hidden in this is also DBSTORE.PUB.SYS, if you are storing a data base that moves to several machines and these machines are a mixture of MPE V/MPE XL CPUs you will want to use the INFO parm TRANSPORT option.

Application Programming

The area with the most press has definitely been that of the CM versus NM issue. Application Programmers will have added complexity than what they have experienced on MPE V/E machines. But a very complete and mature set of tools exist to handle just about any mode of execution that is encountered.

First of all, compatibility mode works as advertised. Simply store your program files on your classic machine and restore them on to your HPPA machine, then run. If porting TurboImage data bases at the same time please remember to disable ILR first. After the data base is restored ILR is not needed as MPE

XL's Transaction Management takes care of ILR functionality. Back to CM, in two years of use at our shop the only problems we have experienced with CM are that of privileged mode CM programs. All other programs tested or in use now in our shop have worked flawlessly. Performance characteristics are surprisingly good. Our shop has seen up to 2x's improvement in some CM applications. While this will obviously vary from site to site, CM performance has yet to be an issue with us.

In between CM and NM is the "pseudo NM" mode of program execution. The Object Code Translator (OCT) is the tool that allows for this mode. OCT, FOS provided, translates CM program and SL files to a pseudo native mode code. An OCT modified program can run faster than CM based programs. A drawback to OCT is the disc space that the translated files take up. Program sizes can grow an order of magnitude. All in all, OCT is a fast and simple way to improve the speed of your CM programs. Experiment with specific programs to see if OCT provides you with increased throughput.

Finally there is native mode. Using the proper compiler command file the switch to NM is not that difficult. COBOLII was our shop's main language tested with NM. The only modifications we had were on old COBOL program files. All others were pretty much recompile and run! Other languages may require some attention to $CONTROL statements to maintain proper data alignment. In compiling to NM remember that you are locking yourself into the HPPA architecture. If your location has several HPPA machines this is not a problem. However if your shop is in a MPE/MPE XL environment be cognizant of the support issues regarding application run time choices. It is far easier to support CM across MPE V/MPE XL families than to have exception NM support. Our shop has used our S/950 in integration testing of major software packages before loading them on to our classic machines. Our CM tests have completed without incidence or any major obstacles. HP has done a fairly complete job in assuring that migration is not an issue. Modes of operation (CM, OCT, or NM) are the choice of the user. All have advantages, all have disadvantages. The best decision is made by the local team comprised of the application programming staff. Tools here provide you with quality choices whatever the mode of operation may be.

In addition to the CM/NM areas a full complement of utilities exist to support the compilers found on the MPE XL machine. If CM is the chosen mode you will find a CM Segmenter to maintain your SL and USL files. On the NM side is the powerful and flexible LinkEditor tool to handle linking and management of your native mode object modules. DEBUG is combined into one program that has a CM mode as well as a NM mode of operation. Users of Debug will be pleased with the increased functionality of the product. It has a Windows like operation that allows a user to define variables which can be set in partitions of the window. Another tool is the the Switch Assist Tool (SWAT) which allows a user to create switch stubs for a NM program that needs to call a CM subroutine. With a View driven interface this tool allows a native mode programer to create the source code needed by a NM program so that a CM subroutine can be called.

The migration from classic to HPPA can be as challenging as you want it to be. A full native mode migration will require recompilation and testing. A pure compatibility mode migration can be as simple as Store/Restore and Run. Obviously time and support issues dictate what path is chosen. CM is the preferred mode in our shop at this time. This is due to the fact that we support several CPUs, the majority of which at this time are classic machines. Yet CM on MPE XL has proven totally acceptable in our experiences. Positive surprizes such as improved performance were exiting to see. What migration path a shop decides upon in the application area is up to that shop's staff. The issues of time, support, performance all need to be resolved satisfactorily. What ever path is chosen, the tools needed are present and waiting to be used.

Conclusion

MPE XL is here, it is functional, and it is powerful. The product provides the tools necessary to smoothly, professionally, and transparently move a HPPA machine into a user environment. Focusing time to learn the nuances of the Operations, System Management, and Application Programming areas will insure a successful migration. Take advantage of the resources available to your site. Attend and participate in Hewlett-Packard classes to gain valuable experience and exposure to MPE XL. Prepare in advance on your MPE V/E machine by using the tools such as Run Time Event Logging (RTEL) and Object Code Analyzer (OCA) to pre-identify any migration issues. Order manual sets ahead of time to get a feel for the activities that will take place. Consult with your SR and SE on migration issues, bring them in to the planning of your migration. Work together so that all issues are identified and resolved in a satisfactory manner. Performing these steps will guarantee the success of the integration of a HPPA machine into your shop. We did it and so can you!

COOPERATIVE COMPUTING - A BRIDGE TO THE FUTURE

Pamela Brown
Hewlett-Packard Co.
Office Systems Division
8010 Foothills Blvd.
Roseville, CA 95678

"Cooperative Computing - A Bridge To The Future"? The future? The future of what? What is "cooperative computing"?

In its broadest sense, cooperative computing implies a systems architecture and a communication infrastructure that permits the sharing of organizational information and applications in a multiple CPU network - PC, minicomputer and mainframe.

A <u>cooperative computing</u> environment can provide the essential building blocks for strategic business systems. It is these systems that will carry businesses <u>into the future</u> with increased effectiveness and a competitive advantage.

It is no secret that advancing information technology affects products, processes, companies, and industries. Firms implement new technology for a variety of reasons. Until recently, most implementation has been in support of new processes for producing or distributing products or to more efficiently support the administrative requirements of running the day-to-day business. Consequently, most implementation planning has typically focused on a specific activity of the business. Often too little thought or planning is given as to how that technology fits or "cooperates" with the technology of the rest of the organization.

More and more managers are beginning to assess the current and potential impact of information technology on their businesses. They are beginning to understand the strategic implications of new technology and how information can be used to improve their businesses competitive positions. As this understanding grows, managers will be able to strategically focus their information resources on the firm's critical success factors.

As managers plan strategically for information technology, the relative priority of investments in information technology may change. For example, increasing the efficiency of a payroll system may not be the investment that yields the highest return. It may be that solving customer service problems with a new on-line order entry system is a more strategic investment for a firm. This new system may cut order-entry costs and provide more flexibility to customers in both the time and process required to place the order. The system has the potential to provide the firm with a significant competitive advantage over rivals. The added value for the customer combined with reduced order processing costs and a potential increase in sales can assure the firm of future success in the marketplace. Obviously, a more efficient payroll system may be very important to administrative efficiency and reduced administrative cost but it does not provide a competitive advantage so necessary to the continued viability of the firm.

The purpose of this document is to describe how a strategic planning process for business information systems can help assure a firm of future business growth. This will be accomplished by:

1. Offering some examples of strategic systems built with today's technology and identifying the key business success factors that these systems addressed.

and

2. Discussing the strategic planning process and how MIS managers need to look at current information capabilities as "building blocks" or "platforms" providing technological growth paths for cooperative computing environments.

EXAMPLES OF STRATEGIC IMPLEMENTATION PLANNING

A Decision Support System For Gas Allocation

The managers for a natural gas commodity trading firm recognized the need to analyze the gas industry to determine the key factors for the success of a business in that industry. They understood that a competitive advantage in that business required that the most current information on market supplies and market demands be quickly analyzed and assembled to provide a timely bid or proposal for the supply or sale of natural gas. Having gained this knowledge about the external competitive environment, they then looked internally, at the organization, to study the current flow of information and identify those areas where improved access to information could provide the most significant benefits.

They looked for areas where there was guesswork caused by lack of information. They also looked at areas where information, currently available, could be more quickly disseminated throughout the organization to provide more timely analysis for the allocation of gas between suppliers and buyers.

The allocation process was done once a month. It required a thorough analysis of the gas supply available, the transportation or logistical requirements, and the market needs. Decision makers from throughout the U.S. and Canada met together one week before the allocation deadline to compare supply and demand spreadsheets and determine how the gas would be allocated at the end of the week.

Equipped with a thorough understanding of the business need and the process flow, they then searched for the information technology that would allow the firm to more efficiently accomplish the allocation process.

They chose Hewlett-Packard's Business System Plus software to provide not only the information access, analysis and file sharing capabilities but also to provide the LAN environment needed for the fast exchange of information. The PC was viewed to be essential to allow individuals to process the information according to their needs yet be able to quickly share this information with other decision-makers throughout the organization.

With the ability to "share up-to-the-minute information on a moment-to-moment basis" decision-makers no longer need to meet together. Using information technology in a cooperative manner, decision-makers can finalize the allocation process within minutes of the deadline. By waiting until the last minute to make the deals with their suppliers they are able to negotiate better prices on their gas purchases. Needy buyers, waiting to be notified of an allocation, may be willing to pay slightly more to assure themselves of the necessary supply of gas. The advantage to the firm is a wider profit margin on their brokerage activities. This is what provides this firm with a competitive advantage.

Focusing on just one area of the business has had a major effect on this firms competitive position. These managers have found the key to success for the firm is through process improvement. To assure the firm's future success they will continue to look for ways to create substantial and sustainable competitive advantages through the deployment of information technology, much of which is available today.

A Business Information System for Hewlett-Packard's Information Systems Group

Even computer firms, like Hewlett-Packard, which are in the information business, need to step back from the fast-paced implementation of technology to study their fundamental business needs.

Managers in HP's Information Systems Group found that answers to basic questions about the group's divisional sales results were often not easy to obtain in a timely manner for decision-making. In fact, given the size, complexity, and international scope of the order entry process, much of the information was not available until several days after the month end close of business. The information, though available, was typically widely dispersed throughout the organization, held in a variety of formats on multiple machines. The business analyst, trying to respond to management needs, found it necessary to gather data from a wide variety of sources and consolidate that data locally to perform the necessary analysis.

Recognizing a fundamental business need for managers to get faster, more timely access to sales information, the group controller and the MIS manager tracked the information that was most frequently required by the various management levels within the group. It became readily apparent that the higher levels of management simply wanted to track some key business metrics in order to "manage by exception". Exceptions would prompt them to ask questions of the responsible lower level managers. These lower level managers, in turn, would require more extensive access to information in order to respond to their managers needs.

After thoroughly understanding the source and flow of the product information throughout the organization's information systems, the MIS manager was able to identify the information technology required to meet the business needs of the various levels of management within the group.

The uppermost levels of management desired a daily, graphical "snapshot" of their key measures for success. They wanted to be able to note the exceptions, understand the reasons for the exceptions, and take the appropriate actions required to better manage their businesses on a day-to-day basis rather than waiting until the end of the month. "Time to action" is a critical success factor.

Examples of a few of the key measures they required are:

- Actual orders versus targeted orders for the business unit
- Actual orders versus targeted orders by region (domestic and international)
- Major product actual orders versus targeted orders
- Trade discounts by major product

You can imagine the kinds of questions that might be asked when exceptions are noted in these measures.

To meet the needs of the upper level managers, the MIS manager implemented a "push button reporting system". Using the command file capabilities of HP's Information Access and Graphics Gallery products, he was able to access information from a variety of computing resources and automate the process of presenting this business information in a graphical PC format. This information is distributed to upper level managers throughout the organization, including the European operations, using the electronic mail capabilities of Business System Plus. Managers simply select one key to automatically view a graphical representation of the daily sales information. This timely access to the information they need allows them to respond quickly to any aberrations in their business.

Lower level managers demanded not only the same graphical reporting capability but also access to the supporting detail information. They wanted to be in a position to anticipate their managers questions, gather and analyze the necessary information and be prepared to proactively (versus reactively) respond to critical needs in the business environment.

The strategic implementation of the integrated PC and HP 3000 computing capabilities of Business System Plus (information distribution, information access, shared resources and personal applications, such as Graphics Gallery) enables Information Systems Group managers to more effectively manage their products, their product lines, and their businesses. The ability to take immediate action in response to aberrations in key business success factors can improve the the entire group's overall performance and contribution to profit.

<u>A Market Sales and Service System For a Consumer Goods Manufacturer</u>

Imagine having 10,000 customers and only 350 sales representatives! This firm is always looking for ways to increase the productivity of these sales representatives while improving the efficiency and quality of the order entry process. The key success factor in this business is customer satisfaction through prompt response to customer needs and timely, accurate turnaround time from order entry to customer delivery.

The vice president of MIS has studied the sales process to ascertain where information technology could not only improve the process but also offer the firm a competitive advantage over rival firms. He found that sales representatives are not spending the entire day calling on customers. In order to keep up with the administrative detail, sales reps return to their offices in the middle of the afternoon, often spending more than an hour on the telephone with the firm's order processing department. Any orders that require immediate input are transmitted over the phone. If customers had questions that the sales rep could not immediately answer, such as product availability, this is also obtained over the phone from the order processing personnel. A follow-up call to the customer is then required to provide the necessary information. All other orders taken during the day are summarized on a daily order form and mailed in for subsequent data entry by order processing personnel.

It was very apparent to the vice president that a significant amount of information is often being handled by multiple people before being processed. Duplication of effort occurs all through the process from the point of the customer's initial order to the servicing of the sales rep requests for information. In many instances customers are waiting several days for a sales rep to call on them to take their orders. These are frequently the instances where the sales rep is required to provide immediate order input via the telephone. At each step in the process there exists many opportunities for the wrong information to be entered into the firm's information systems. Inaccurate order information leads to shipment of the wrong products. This results in a "snowballing" of corrections. The correct information has to be reentered; credits for returned merchandise have to be entered; inventory has to be adjusted; customer rebillings have to be issued; etc..

The rest of the industry has the same problem. The vice president knows that the strategic implementation of information technology could offer his firm a competitive advantage over their rivals. The advantage might be short-lived but it will take awhile for the competitors to catch up thus giving this firm a chance to grab valuable market share.

This firm plans to put portable computers in the hands of their sales force. The portables will be equipped with a cooperative processing order entry application that provides access to and update of the central database. The sales reps will enter the order information while at the customer site, verifying the accuracy of the data entered with the customer. Larger customers will be provided with their own personal computers and access to on-line order entry. Using HP's Cooperative Services, a significant amount of product information can be downloaded to the portables so that sales reps and customers can answer the majority of the most frequently asked product questions. At the end of the day the sales reps will upload the order information to the central database. If necessary they could dial-in to this central system from their customers site to immediately enter the orders or obtain such information as product availability from the central database.

Through the implementation of this integrated order entry system, sales reps will be relieved of the daily paperwork. Their job satisfaction should increase because they will be able to concentrate their efforts on what they do best - selling. The elimination of the paperwork and most of the telephone calls should allow for more time to call on current customers and cultivate new customers.

Not only is there potential for an increased level of sales but overall organizational effectiveness will be improved. The need for the middle layer of order entry personnel will effectively be eliminated. The more accurate entry of information into the system will eliminate the need for numerous corrections. The ability of the portable and the central system to cooperatively process the order information will provide for the most effective use of the firm's computing resources.

More importantly, a competitive advantage may be obtained. Customers will be serviced in a more timely and accurate manner. Larger customers will have even more flexibility in the timing and entry of their orders. The result should be an increased level of customer satisfaction, a critical success factor for this industry that can assure this firm of continued growth in the future.

GROWTH PATHS TO COOPERATIVE COMPUTING ENVIRONMENTS

In each of the above examples, management assessed the key success factors for their businesses and identified how information technology could impact their competitive position. They strategically planned for the integrated information systems that would meet their business needs.

This growing need for truly integrated information systems is due to various technological, economic and organizational factors such as:

- The need for fast, reliable information exchange in response to rapidly changing markets, products, and services.

- The evolution of guidelines, standards, and protocols.

- The penetration of information systems into internal business processes.

- The increasing technical quality and capability of information systems technology.

- The use of information systems technology to distinguish a product and/or a company.

How can an MIS manager provide these types of solutions with limited resources and constrained budgets?

Today's environment seems to require an ever-increasing sophistication in terms of software, hardware, and networking technology. Certainly, none of the examples described earlier could have been developed in reasonable timeframes if MIS had started from scratch with basic Cobol programming.

The answer, clearly, is to start with higher-level building blocks to leverage cooperative computing "platforms".

Cooperative Computing Platforms

Platforms, in this context, can be used to describe an architectural framework upon which information systems can be built. The ultimate goal of the platforms is to provide the structure for a cooperative computing environment to support the information systems. The information systems themselves are the solution.

Hewlett-Packard's goal is to provide the fundamental foundation through:

- A Scalable RISC Computer Systems Family
- Easy to Use Integrated Workstations
- Industry Standard Networking
- Industry Leading Support

Industry standard networking will provide the link between HP computer systems and software and the network of heterogeneous systems and software present in so many organizations. To enable a platform to support cooperative computing and coordinated work, this network linkage is key to the distribution of information and computing throughout the organization.

HP's NewWave interface architecture, focusing on business tasks rather than application tools, will provide the transparent, object-oriented environment necessary for these easy to use workstations.

Upon this foundation, HP is committed to offering building blocks, such as Business System Plus, to fulfill the fundamental needs for integrated individual, workgroup, and organization-wide communication and computing. These building blocks are necessary to construct information systems which can take advantage of a "cooperative computing" environment to support the firms strategic plans.

Today, HP's Business System Plus integrates the powerful information access, analysis, sharing and distribution capabilities needed to build multi-purpose information systems such as the example described above for gas allocation. By providing intrinsic access to data, files and services through products such as Cooperative Services and the HPDeskmanager intrinsics, these multi-purpose information systems can become highly integrated single-purpose transaction systems similar to the order entry system also described above. These same systems can be further customized by adding third party product pieces to provide truly competitive information system solutions. These building blocks can provide the foundation for comprehensive Decision Support Systems, Executive Information Systems, Medical Analysis Systems, etc..

In summary, HP will continue to provide more and more building blocks designed to be integrated in a truly cooperative computing environment. HP itself, our customers and our third parties will be able to use these building blocks to provide the strategic information systems necessary to achieve their businesses critical success factors.

MIS managers need to view the office functionality available from HP and third parties as essential building blocks, or system development platforms, for these future strategic systems. For example:

- The need to view a capability like HPDeskManager, not as an electronic mail system for only person-to-person messages, but as an information transport system to tie together operational systems.

- The need to view data access capabilities, PC spreadsheets, and graphics tools all as component building blocks of strategic operational systems.

As business managers grow in their level of sophistication in identifying key success factors and planning for the strategic implementation of information technology, it will be increasingly important that these building blocks be in place. After all, time to action is critical to firms desiring to obtain a competitive advantage and it is a competitive advantage that will ensure the firm a "bridge to the future".

ONE SOURCE, MANY MACHINES: APPLICATION DEVELOPMENT USING HP PASCAL

Jean H. Danver
Hewlett Packard Company
Cupertino, California

Overview:

This paper presents two topics. The first is migrating a large application from MPE/V to MPE/XL in such a way that the resulting source is shared for both systems (and HP-UX as well). The second is information on migrating Pascal applications in general to MPE/XL. The HP Pascal compilers: Pascal/V, HP Pascal/XL and HP Pascal/HP-UX were originally a set of compilers for the Classic 3000 and cross compilers based on those compilers for experimental computers. This software was ported to MPE/XL and Series 800/HP-UX as native Pascal compilers. How this was done and the final result is explained. The source changes required for migration will be pointed out, as well as how they differ from what a user would have to do today. Since a compiler is an example of only one type of application, it does not have all the common migration problems. So, the compiler features supplied to aid migration of other types of applications are discussed. Migration of applications using other common subsytems such as View and Image is not covered. General tips for the migration of Pascal applications are given throughout and a list of publications that can help is included.

The Challenge

It was our job to produce a pair of Pascal compilers for HP-PA that was compatible across operating systems and minimized migration effort from Pascal/V. We were to do this before there was any hardware or operating systems available, and at the same time we had to provide development tools for the projects using HP Pascal. This included all of MPE/XL, SQL-based database products, most data communications software and parts of several compilers. When this started we had the Pascal/V compiler, an internal version of Pascal/V known as MODCAL and a MODCAL cross compiler for a canceled computer project known internally as Vision. The challenge was to pull all this source together and end up with a shared source system that produced three compiler products: Pascal/V, HP Pascal/XL and HP Pascal/HP-UX. The only way this could be done in the time-frame needed was to port the front end of the Pascal/V compiler.

The Result

Four compilers (Pascal/V, HP Pascal/XL, HP Pascal/HP-UX, and MODCAL/3000) emerge from over 250,000 lines of Pascal source code. They are targeted for two architectures and three operating systems and all share the same front end source. The source is maintained on an HP-3000 Series 68. Each compiler is created on the target machine by moving the source over a network for compilation and testing.

Logically, the source is organized in a hierarchy of directories. The main directory is called official (contains all the official source). This is actually an account called official. There are three logical directories for sources known as fe (front end), pa (precision architecture) and 3k (classic 3000). Each of these logical directories contain four other logical directories. They are proc (procedures), decl (declarations), ext, (external procedure declarations) and ob (outer block). In reality, the logical directories are represented by groups under the official account. They are named procfe, procpa, proc3k, declfe etc. The actual source files are in these groups. The fe groups contain source that is shared with all the compilers. The source in the pa and 3k groups are primarily code generation routines that are aimed at a particular architecture. The files in the ob groups are the ones actually compiled. They include the files in the other groups during the compilation. As a rule they are set up as job files. The latest usl files used to create compilers for testing are also there. Source is managed using an internal source management tool. The actual source files are named after the procedure they contain (every procedure is in a separate file). Having all the files in one account facilitates dual development with HP-UX and the requirements of the source management system. It also makes it alot easier to replicate the compiler source somewhere else, if needed. All the includes are done without reference to account name, for example, which allows putting the source in a different account very easy. A picture of the source organization is in Figure 1.

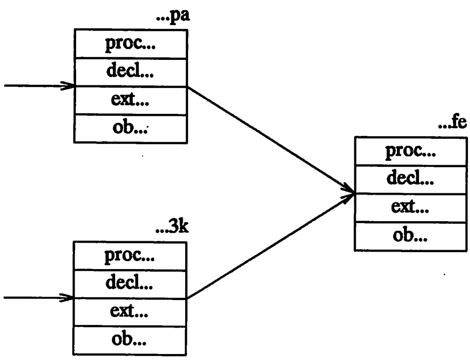


Figure 1

Using a Classic 3000 development environment is not the only logical choice. An MPE/XL machine could be used. Cross development for the Classic 3000 can be done by using compatibility mode. Object files between MPE/XL and the Series 800 HP-UX are compatible, so the HP-UX compiler front end could be compiled on MPE/XL and transported over to the Series 800 for linking.

Because part of testing for the Pascal compiler is to compile itself on the target machine with the target OS, source and tests have to be moved anyway. So, each compiler is routinely produced on the target machine. All the machines are connected in a network. The engineers develop on whichever one they choose. HP-UX is not suitable for base development because of the lack of compatibility mode and 3000 format floating point emulation. Though debugging and development that does not involve those features is frequently done under HP-UX because some of the engineers prefer it (especially those who learned UNIX* in school).

How We Did It

Changes Required for Migration & Source Sharing

A. Pascal Dependencies

Language features presented the least of our problems. They centered around one issue -- dependencies on the Pascal/V packing algorithm. The compiler did not have too many and

* UNIX is a trademark of Bell Laboratories.

they were pretty well isolated.

There were some debugging routines that displayed pointers values as integer values. This was done with a tagless variant record that overlayed a 16 bit integer with a pointer. The problem of pointer overlay solved itself. The type declaration Smallinteger: -32768..32767 is allocated 16 bits in Pascal/V and 32 bits in Pascal/XL, the same sizes and alignment as the respective pointers. So, we were still able to display the pointers. If we had been doing variant record pointer arithmetic tricks, changes would have been required. Heap pointers on Pascal/V are 16 bit word offsets and on HP-PA they are 32 bit byte offsets. Identical manipulation does not get the same offset changes.

HP Pascal has a feature known as structured constants. This is the ability to declare a record, array, set or string constant. The feature is implemented by building the array or record constant at compile time in a buffer area. Implementation requires moving arbitrarily sized objects on arbitrary bit boundaries to arbitrary bit boundaries. This was accomplished by creating overlays using tagless variant records. Needless to say, the layout came out differently in Pascal/XL. What we did was to create two record definitions that had the same packing in both compilers. The Pascal/V one remained unchanged. The Pascal/XL one was the same structure PACKED. ShortInt was used to obtain a 16 bit integer. Conditional compilation ($IF..$ELSE..$ENDIF) was used to put the changes in source.

It is usually not difficult to create two record structures with the same packing on each implementation. The declaration may be a little different. For example, just making a structure PACKED on XL will frequently be the same as unpacked on Pascal/V. We found that using conditional compilation to cause differences in declaration was much better than changing algorithms to manipulate things. A very useful compiler option to use in the checking out of packing is $TABLES. In both Pascal/V and Pascal/XL it will print out the structure layouts. The output can be used to verify that the layouts are the same.

This solution points out several things:

- Use the predefined type, ShortInt, to get a 16 bit integer in XL. Be sure that it is not declared in the XL program, as this declaration will override the predefined type and will likely not be 16 bits.
- Manipulating declarations to get the same packing as in Pascal/V is usually preferable to changing algorithms. It is also preferable to using $HP3000_16. Why this is the case will be explained below.
- Conditional compilation is very useful for coding small differences when the source is being shared.
- Small integers with negative ranges are allocated 32 bits on XL and 16 bits on V, so pointer overlays can share the same source, as long as the overlay is for display or comparison purposes only (and extended addresses are not being used).
- $TABLES can be used to check the layout of declarations.

B. Bug Fixes

Believe it or not, there were bugs that had to be fixed that did not show up on the Classic 3000. These bugs were uninitialized variables. Garbage on the Classic 3000 tends to be zeros. Zero frequently is fine for an initial value. On HP-PA garbage really is garbage. So, some pretty confusing bugs appeared. Even though we were aware of this, having a feature work on MPE/V and not on MPE/XL or on one version of XL and not another almost always lead us to suspect the OS. (Customers suspect the compiler). But more often than not, it was an uninitialized variable on our part.

Users run into this problem with string variables very often. They might use Strwrite on a string, for example, but forget to assign a null string to it before hand. Strwrite will update the length, if it is expanded, so everything works fine on the Classic 3000. On XL, there may be some huge number for the garbage length which will result in an abort as soon as anything is done to cause a range check. Whenever you have a program abort on a string when it doesn't abort on the Classic 3000, look for uninitialized string variables. These problems come up with several of the string routines. Strmove and StrAppend are common routines that are incorrectly used to initialize variables.

String variables were rarely our problem. We were too aware of that one. Our uninitialized variable problems tended to be fields in records that a called procedure expected to have a certain value and the calling procedure did not fill in. The zero on the stack was the correct value on the Classic 3000.

These are the random porting problems that cause the biggest headaches. My advice is to learn NM-Debug or XDB, and have faith that it is your problem and suspect uninitialized variables.

It is important also to develop programs that do not reply on undetected range errors requiring $RANGE OFF. One great advantage of Pascal is that random garbage usually results in a range error of some sort, so the problem is detected fairly soon. Do all your development and testing with $RANGE ON.

C. Operating System Dependencies

Extra Data Segments

The operating system dependency relied on the most, which did not map onto XL, was extra data segments. The Pascal/V compiler makes heavy use of extra data segments for reducing stack use. For example, all symbol names and constants are kept in extra data segments. Using extra data segments to save space makes no sense on HP-PA. There is lots of space. The logical thing was to switch to heap use. However, lots of code in the 3000 compiler was written with extra data segments in mind. We did not want to rewrite everything and, more importantly, we wanted to share source. To accommodate this, the extra data segment accessing routines were changed to do heap access instead.

This turned out to be trickier than first expected. It involved a mapping of a segment of 16 bit words onto a heap variable of 32 bit words. It was decided to keep all "extra data segment" addresses in terms of 16 bit offsets. This was not exactly intuitive for HP-PA, but did result in shared algorithms. All the "messy" stuff was encapsulated in a few routines with names like XSegPut and XSegGet. They handled all the heap addressing on HP-PA and extra data segment addressing on MPE/V.

In addition to space availability, efficiency considerations also contribute to differences. For example, the structured constants part of the compiler had a buffer scheme in the stack to reduce calls to the extra data segment routines. This made no sense in the HP-PA compiler. It amounted to double moving. The 'extra data segment' was heap. It was much more efficient to go there directly. This issue caused a rewrite of some of the structured constant building algorithms to make them more general. The bottom line is that programs should not only have machine dependencies isolated because you may have to port some day, but also one should isolate in the same way (or resist) those 'efficiency' algorithms based on your particular machine.

There will always be ways to make things run faster on a particular machine. You should be careful to only tune those things which need to be tuned. One guideline is to tune only those things which are on the most frequently used path. Another is to only make machine dependent improvements only when there is a measurable performance improvement of greater than some percentage.

Command Interpreter

All job streams had to be recreated, just like yours, because of differences in running jobs, libraries and linking and loading programs.

The compatibility between the MPE/V CI and the MPE/XL CI permitted a phased migration of the development environment. Initially, the environment was ported intact with only necessary changes being made, e.g. using the Linkeditor versus the Segmenter.

One interesting quirk of which to beware is that new 'reserved words' have been added to the MPE/XL CI. A UDC named DO which was brought over caused considerable confusion whenever it was executed until it was realized that it is one of the CI's new 'editing' commands. *Unlike their counterparts in the MPE/V CI, these commands cannot be superseded by a UDC.* As familiarity with MPE/XL increased, the development scripts were altered to take advantage of two of the MPE/XL CI's most powerful features: **command files and environment variables.** These are not backward compatible, but can be used for compatibility mode compilation.

The interface between our command files and environment variables is similar to that of modules and imports or procedures and parameters. As many operations as possible (compiling, linking, error reporting, etc.) were modularized into command files. Since as many as fourteen varieties of the self-compilation script exist, this modularization greatly decreased the time necessary to modify any part of the compiler generation process.

Information for the command files is provided via environment variables.

Working from a basic self-compilation template, the environment variables are initialized to provide a specific compilation environment (destination of the resulting object files, optimization, symbolic debugging, etc.) Among the environment variable capabilities which we have found most useful are:

- The predefined environment variables, particularly those specifying the invoking user (hpuser) and group (hpgroup). This allows a single command file to be tailored to behave differently dependent on who is invoking it. Others which are especially useful are hpdatef, hptimef, hpusercapf, hpwaitjobs, and hpautocont.
- The ability to set environment variables to string values. This allows file names, e.g. the compiler to be used, to be passed to command files. When combined with the ability to parse and compare these strings, powerful preprocessing may be performed. For example, we were able to produce an environment that looked like the scripts we used in HP-UX. This made the operating environments look the same so that the software engineers were not having to do an environment switch in their heads all the time.

Intrinsics

Some intrinsics changed from MPE/V to MPE/XL. Most intrinsic changes are hidden by the intrinsic mechanism. All intrinsics used by the compilers were declared as intrinsic, so most changes were automatically taken care of. Some intrinsics did cause minor changes. These were those for message catalogs, CREATEPROCESS, and traps. The compiler calls intrinsics, such as FOPEN, that have 16 bit returns in XL. The predefine, ShortInt, was used here.

Both of these were isolated by conditional compilation. Our use of CREATEPROCESS was based on preserving stack space on the Series/V. The compiler actually creates a separate process to do the cross reference. This use was removed from the XL compiler.

The Pascal compiler tries to do some reasonable recovery if a trap occurs while it is running. The code was added to use the new set of intrinsics. The old MPE/V intrinsic, XLIBTRAP, does work. The new intrinsics, ARITRAP, HPENBLTRAP, and XARITRAP, are on HP-UX as well as MPE/XL, which fit our purposes very well for shared source with HP-UX and they give better control of trap handling in general. As a result they were used in the HP-PA compilers. They are not backward compatible with MPE/V. These routines are all documented in the *HP Pascal Programmer's Guide.*

The format for intrinsic files changed from Classic 3000 to HP-PA. This is not a concern for user programs since compilers are the only applications that access intrinsic files. Some users have their own intrinsic files. These must be converted to the new format. How to do this is explained in the *HP Pascal Programmer's Guide*. The compiler had to change the part of the compiler that accessed intrinsic files because of the new format.

Pascal/XL builds intrinsic files. In fact, the only way to build intrinsic files on XL is to use the Pascal compiler. As a result of this, the compiler group frequently became involved with the production of the system intrinsic file for XL. Using the 'wrong' or out of date system intrinsic file was a frequent source of problems.

Architecture

Sixteen bit arithmetic causes hardware traps on the Classic 3000 when an operation's result is greater than 16 bits. The compiler used this fact to optimize code generation. Rather than doing arithmetic in 32 bits for a binary expression whose result had to be 16 bits we used 16 bit arithmetic, allowing a hardware trap to catch range errors. This could not be done on HP-PA, since all arithmetic is done in 32 bit registers and no trap would occur. So, in the HP-PA compilers we needed to generate range checking code to catch the range error. Where range checking code is to be emitted is determined during semantics processing. During code generation we either generated checking code or not depending on the semantic result. Our range checking algorithms were dependent on the Series/V architecture. These were re-written in a more general manner and some of the processing was delayed until code generation, which is not a shared part of the compiler.

There were internal base type representations of numbers based on whether they were 16 or 32 bit arithmetic. This was determined during semantics. Here was another case where the algorithm was dependent on the Series/V architecture. There was no need for the 16 bit representation and it caused some amount of confusion, since what we did at code generation was affected by the representation of the number. We also experience some problems with I/O in the run time library which keyed off the internal representation of the number. User programs should not experience these kind of difficulties, but you never can be sure. Whenever you are doing anything that has a data dependency, there might be migration issue.

D. Special Feature Dependencies

There are some features of the Pascal compilers that turned out to be essential for source sharing. The most important of these has been mentioned several times. That is the conditional compilation mechanism. Obviously, one does not want to duplicate source for a minor change. It took some experimentation to get this feature right for large application development. $SET options which give values to the conditional variables must appear before the PROGRAM header in a compilation. All variables must be given values (no defaults) and they cannot be changed later on. These may sound restrictive, at first, but no misspelled variables get default values and there are no behind-ones-back changes later in the compile. We did not want bugs from compiling or not compiling a piece of source by mistake.

Also essential is the nesting of $IFs. That is, to be able to put a $IF inside of a $IF:

```
$IF 'HP-PA'$
....
   $IF 'XL'$
   ....
   $ELSE$
   ....
   $ENDIF$
....
$ENDIF.
```

This may not seem so at first glance, after all, there is only the 'Classic 3000' and HP-PA. Well, what about HP-UX. At the high level we have machines (3000 and HP-PA), which may contain operating system conditions (UX and XL).

Conditional compilation also allows conditional development. For example,

```
$IF 'new_xyz_feature'$
 ....
$ENDIF$
```

The production compilation sets 'new_xyz_feature' to FALSE. When it is debugged, it is set to TRUE or the $IF was removed from the code.

We use two other features of the compiler to make self-compilation possible. We need to have source on the target machine for a self compilation test. Different operating systems have different file naming rules. This presents a problem for source files included with $INCLUDE, of which Pascal has thousands, since the OS specific file name is specified in the include. Filename.group.account is not what UNIX * expects to see. A compiler option, $Convert_MPE_Names, was developed to convert filename.group.account to ../account/group/filename. This enabled an accounting structure to be set up in HP-UX that would compile the same source. The problem does not arise between 'Classic 3000' and XL. $Convert_MPE_Names is being released in HP Pascal/HP-UX.

Another useful internal option is one that logs to a file all the files that are included in a set of compilations. The result is a list of the files that need to be moved across the network. A cross compile is done with the conditional compilation flags set for the target machine, the sole purpose of which is to get the file list. This is also being released in future compilers with a yet-to-be-determined name.

War Stories, or Things You Don't Have to Worry About

A. The Compiler or the Operating System or the Computer

---

* UNIX is a trademark of Bell Laboratories

Developing an application for an operating system and a computer that did not exist certainly had its challenges. It would take a book to describe the process, the things that did not work, and the 'temporary' solutions that were required. A few things will be mentioned to give you the flavor of what went on.

A native compiler requires a native operating system on, at least, prototype hardware. There were simulators which were hopelessly slow and emulators which were in short supply and not real fast either. We were able to use HP-UX for development. Since the source was shared, work done on the compiler applied to both compilers. HP Pascal was a working native compiler on the 800 awhile before XL was ready to run it. When XL was ready, we were ready, too. This greatly speeded up XL development.

MPE/XL is written in Pascal/XL. It often was not easy to determine if the OS had a bug or the compiler had a bug. When there was a doubt, the compiler on HP-UX could sometimes be used to make the determination. That did not always work. There were many evenings spent with Pascal and MPE/XL engineers huddled together trying to determine what was going on. Both compiler and OS work might stop when this happened. The thought of all that engineering talent going to waste was pretty motivating to solving problems. NMDebug was not yet working well. Problem areas tended to be XL source management, a reliance on bugs that got fixed in a new version, uninitialized variables and too many other things to remember.

B. Rollovers

A rollover is when a change was being made, usually to generated code, that is incompatible with previously compiled code and requires every piece of code to be recompiled. That doesn't sound so hard. You just get a new compiler and recompile your source, right? Right for a compiler user, but not the compiler. It does not make the user happy and it is time consuming, but it is straightforward. Well, there are real pump-priming problems with rollovers. We did two major ones during HP-PA development. These were changing the code generation for procedure calls and changing the convention for the external names of procedures (the link names were changed from upper to lower case).

1. New External Naming

In the beginning the operating system runs in the old naming convention and expects code in the old naming convention. To prime the pump, a Pascal compiler is produced that runs in old naming and produces new naming convention. The implications of this is that the Pascal project had to produce and maintain a variety of compilers for awhile:

a. old/old - For the users that were still using the old operating system and compilers that run on them. They tended to be alpha test sites.

b. old/new - For the users that had the old operating system and needed to compile their code to run on the new.

c. new/old - This was for the operating system which expected user code to be in the new names (Pascal is a user program), but needed old names produced because that was what they were still using.

d. new/new - The end result. This was regular users on an operating system expecting new names.

All the assumptions coded-in concerning external names needed to be discovered and removed or changed before the operating system would work. What actually was done was, rather than rolling themselves, the OS modified code so that it required user code to be in new naming, but it ran in old naming. This resulted in rolling the rest of the world faster, but compounded Pascal's version problem. Hundreds of little routines were written called stubs. A stub was a routine in one external name that turned around and called the real routine in another external name. Routines called by users and the OS linked in stubs in either direction depending on the external names of the actual routine. Once the rest of the world was running in new names the operating system switched itself over. The whole process took months. Some funny bugs resulted in the end. For example, a stub for a particular routine was written in new naming that called old naming. Later the routine was converted to new, and a stub was written expecting old and converting to new for operating system use. As a result, our code was linking in a stub that got linked to another stub that reversed the names. Everything worked, even though there was two unneeded stubs. The OS converted, deleted the old to new stub and suddenly we could not load programs anymore. It was quite a surprise, until someone realized what was going on.

2. Procedure Call

The procedure calling convention is the code sequences that are used to cause a procedure call. There has to be agreement in an operating system what these code sequences are in order for things to work. Changing the procedure calling convention was much more complicated than changing the external names. There were blocks of assembly code that were coded in the old convention that had to be recoded and debugged. Debugging each one required all the others called before it to be debugged. The problems could only be discovered serially. The process also required the same flavor of compiler versions mentioned above. The details of this rollover will not be included here. It belongs in a book like *The Soul of a New Machine*. It is the kind of change that can not be made after a product is released.

C. Getting in Your Way When Producing Yourself

Writing a compiler in itself is a bit of a chicken and egg problem. For the most part Pascal/V was used to produce an XL-compatible cross compiler which was used to produce the native compiler. When there was a more stable operating system, the native compiler was used to produce the next native compiler. However, the cross compiler is still used for include log lists, making sure everything compiles (the master source is on the Classic 3000) and debugging. As a rule there is no problem adding a feature to a compiler written in itself once you have a version of the compiler. This works because you do not need the feature to write the code to put it in. The most common problem that arose was destroying the cross compiler because of some interaction of the new feature with an existing feature that did not show up until, well, it was too late to back up. For example, someone would change a global declaration, check all their changes in and then the resulting cross compiler would not work. This left everyone else in a state where they could not build a compiler to check out their changes and, if XL wanted a hot one fixed, the project was in a hot seat.

We would get into what looked like chicken/egg problems when a run time library routine would change its interface. The compiler could not produce itself without lots of intervention. This was not difficult, just detailed, and one had a tendency to not realize it until the compiler failed to produce itself. This is how it works. New source (with the new library interface) would be compiled with the old compiler which needed the old run time library. The resulting compiler ran with the old library and produced source that required the new library. This untested compiler was delivered to the back end project to produce a back end that did the new call. It was also used to compile the source again. That created a front end that required the new routine and produced source that did it as well. It was combined with the new back end and a new library to result in the final compiler.

Later, the run time library routines were made extensible, so this type of roll would not have to be done again. (When compiled with Option Extensible, parameters can be added to routines without affecting existing compiled programs that call the routine.)

Problems We Did Not Have

A. Stack Space Limitations

Pascal programs compiled on the classic 3000 are limited to one 32k byte segment for data space (heap, global data and stack) at run time. XL programs have a much, much larger limit. This problem can interfere with portability, as pointed out in the extra data segment discussion above. It also limits backwards compatibility, as well. All kinds of decisions are traded off. Structures on the Classic 3000 are usually organized to maximize space savings. Giving up a little space in a structure may increase portability, backwards compatibility, simplicity and decrease speed. But, it is not possible when you run out of space. This problem was worked around internally by using an 'extended heap'. This feature increased heap size by using a cache scheme in the stack segment and putting heap overflow into extra data segments. When extended heap is needed things can go pretty slow. However, it greatly increased source sharing between Pascal/V and Pascal/XL.

The Pascal/V compiler itself now runs in extended heap when needed. This removed most compiler limitations from getting in the way of dual development. User programs themselves cannot run in extended heap, so cannot get around this potential compatibility problem.

This feature has not been given to customers because of the performance of applications running in extended heap and the limits on what can be done with an extended heap. Heap addresses are not stack pointers anymore. Any changes of pointer values in a program will not work. TOOLSET does not know how to debug extended heap pointer values. Files cannot be put in an extended heap. However, if this feature is important to you, you should make your needs known to Hewlett Packard.

B. Existing Data

Compilers, as a rule, do not have any existing data files or data bases, so there was no data that had to be converted or exist in two environments. Features were put in the compiler to support user applications with these types of conversion problems. The section below discusses them. Compilers do read files, which could have caused some conversion of formats. However, the files we read were ASCII files, so no conversion was necessary. You should keep this in mind while doing development for dual environments. Use ASCII files or Pascal data files with data packing that has the same layout in both Classic 3000 and HP-PA.

Run-Time Support Changes

Along with the compiler, a run time library is provided with Pascal. In Pascal/V, the run time library handles I/O, heap support, strings, some set manipulations, and some of the predefines, such as Hex, Octal and Binary. There were substantial changes made to the run time library routines for I/O, heap and strings. Set routines were no longer needed. The code generator provided the run time support.

The I/O routines on MPE/V were written in SPL for historical reasons. These were re-written in Pascal.

The heap routines manipulated the 3000 Classic 3000 DL-DB area of the stack. This was rewritten to use HP-PA addressing instead. It also had to be changed to take into consideration HP-PA alignment restrictions for data. This was another case of an architecture dependency.

Strings had a similar problem. The resulting size of some string expressions can not be determined at compile time. This did not present a problem on the Classic 3000 because the data stack could vary in size. This was not true in HP-PA, which requires fixed size frames. Those expression values are now done in the heap.

Compiler Support for Customer Applications

Most of the information here is covered in the *HP Pascal Programmer's Guide* and the *HP Pascal/XL Migration Guide*, and you should use those guides for reference.

Migration was and still is an important part of XL development strategy. Very early in the development cycle there was a task force devoted to drawing up the strategy and the technology that would be developed to achieve migration. The task force recognized that migration was not something that happened overnight. Classic 3000s would be around for years to come. An application may be converted in stages. Therefore, users will need to have parts that ran on both MPE/V and MPE/XL and shared MPE/V data. What this meant for the Pascal/XL compiler was:

1. All features of Pascal/V must be in Pascal/XL
2. Pascal/XL must be able to run in an environment where the data is in Series/V format
3. Pascal/XL must provide a way to enable conversion of Pascal/V data files to Pascal/XL data files

A pair of compiler options and two conversion routines were developed to accomplish these goals. These are $HP3000_16, $HP3000_32, StrConvert and SetConvert. In order to enable programs to co-exist with Pascal/V programs and other applications running under MPE/V the $HP3000_16 options were created. When $HP3000_16 is on data is packed in the same format, when possible, as Pascal/V. This means:

1. all reals are in MPE/V real format
2. strings and sets have the Pascal/V format, which is different than the HP Pascal/XL format
3. types that do not contain files or pointers are sized and aligned the same as Pascal/V
4. all data manipulation assumes MPE/V real numbers and Pascal/V sets and strings

- Accessing Data

  So, now you can interact with Pascal/V data using $HP3000_16. The two exceptions are files and pointers. Native mode pointers are 32 bits. So, structures with pointers will not be laid out the same as in Pascal/V, nor will structures containing files. This ordinarily should not matter. Files are not assignable, so structures with files will not be stored in data files anyplace. Pointer values make no sense, except in a particular invocation of a program. Hence, they rarely get stored as data. If you have a problem here, use a 16 bit integer, such as Shortint, instead. That has the same size and alignment as the Pascal/V pointer.

  It should be pointer out that $HP3000_16 should only be used to manipulate MPE/V data. It was not designed as an alternative packing algorithm. Code generated to manipulate strings, sets and real numbers is not the same as when HP3000_16 is not in effect. You cannot mix routines compiled with HP3000_16 with ones that are not. Some rather strange things may result.

  You can write a program that can manipulate both types of real numbers (IEEE and MPE/V). However, each type must be in different procedures compiled separately. Since we discovered that the compiler could make use of this, it appears that some users programs may also have a need. A sample program is in Exhibit 2 that shows how to do it.

$HP3000_16 does not always need to be used to access Pascal/V or MPE/V data. It is only needed when the data or layout is different. It may be possible to use the same techniques that were discussed under Pascal dependencies. Declarations may be modifiable to create identical layouts. ShortInt can be used to get 16 bit integers. A simple type renaming may be all that you will need.

Data in ASCII files is the same in both environments. So are many simple structures such as integers, char, Packed and unpacked arrays of integer and char. If your external data is of this form, the same program will run in both environments.

- Converting Data

In order to convert an application with existing data, the data may have to be converted as well. To support the conversion of data, the compiler option $HP3000_32 and the routines StrConvert and SetConvert are provided. These, in conjunction, with the system intrinsic, HPFPCONVERT, are all that are needed to convert Pascal/V data files.

$HP3000_32 can only be used when $HP3000_16 is in effect. It will produce a structure that is laid out identically to the HP Pascal/XL packing. Its purpose is strictly to allow programs to be written to convert data files. Strings, sets and real numbers, for example, have the default XL packing and cannot be manipulated in the program. They are not assignment compatible with HP3000_16 strings, sets and real numbers. The conversion routines are used to obtain values for variables of these types.

Strconvert converts a Pascal/V string to a Pascal/XL string and has the form, StrConvert(PascalVstring,PascalXLstring). Setconvert will do the conversion in either direction and has the form SetConvert(VorXLSet, OtherFormatSet). So be careful with SetConvert. You could destroy your data if you get the parameters out of order.

There is a good example of the use of these options for file conversion in the *HP Pascal/XL Migration Guide*. It is repeated in Exhibit 1 with some slight improvements. As you can see, writing a program to convert a data file is quite simple, short and straightforward. The only major complication would be tagless variants. When there is an overlay with a tagless variant, you can not determine what the type of the actual data is. This makes it difficult, to say the least, to convert it.

- Switch Stubs

There are, of course, legitimate exceptions to everything. $HP3000_16 is frequently used in writing routines that call switch stubs. Switch stubs data structures are HP3000_16. Just don't use strings and, if sets have to be used, make sure the layout is the same as Pascal/XL default (this would be the case for unpacked sets that take up multiples of 32 bits in Pascal/V). Files are out of the question. The control blocks are completely different. Use Fnum, if that has to happen.

Conclusion

Source sharing for applications that are targeted for MPE/V and MPE/XL is quite feasible. It offers an opportunity to leverage an implementation investment of the past and the future. If the future includes HP-PA with HP-UX, the opportunity continues. The Pascal project in Hewlett Packard's Computer Language Lab is successfully doing this. The result is a more reliable, compatible compiler in a very short period of development time.

Useful Publications

1. *HP Pascal Programmer's Guide* (31502-90002 or 60006)
2. *HP Pascal/XL Migration Guide* (31502-90004)
3. *Introduction to MPE/XL for MPE/V Programmers* (30367-90005 or 60004)
4. *MPE/V to MPE/XL:Getting Started* (30367-90002 or 60002)
5. *Switch Programming Guide* (32650-90014 or 60030)

Acknowledgements

I wish to thank the members of the HP Pascal group, past and present, for ideas, memories, suggestions and editing.

Exhibit 1

The following program illustrates the migration of a data file from Pascal/V to HP Pascal/XL.

```
$HP3000_16$
PROGRAM Convertfile(file1,file2);

CONST
  HP3000_32bit = 1;
  IEEE_32bit = 3;
  RoundToZero = 1;

TYPE
  Arr1 = ARRAY[1..10] of -32768..32767;  { 20 bytes allocated }
  CMrec =
        RECORD
        f1:char;
        f2:Boolean;
        f3:string[40];          { 44 bytes allocated }
        f4:Arr1;
        f5:real;                {MPE/V representation; 2 byte aligned}
        f6:set of 0..15;        {2 bytes allocated }
        END;
 NMArr1 = $HP3000_32$ ARRAY[1..10] of -32768..32767; {40 bytes allocated}
 NMRec = $HP3000_32$
        RECORD
        f1:char;
        f2:Boolean;
        f3:string[40];          { 48 bytes allocated }
        f4:NMArr1;
        f5:real;               {IEEE representation; 4 byte aligned}
        f6:set of 0..15;        { bytes allocated }
        END;
 file2type = $HP3000_32$ FILE OF NMRec;

VAR
  file1: FILE OF CMRec;
  file2: file2type;
  v1: CMRec;
  v2: NMRec;
  inx: 1..10;
  status : integer;
  except: -32768..32767;

PROCEDURE hpFPConvert; Intrinsic;

BEGIN (*Program Convertfile*)
Reset(file1);
Rewrite(file2);
WHILE NOT Eof(file1) DO
  BEGIN (*Read and Write*)
  Read(file1,v1);
  WITH v1 DO
```

```
    BEGIN (*Assign the components*)
    v2.f1 := f1;
    v2.f2 := f2;
    StrConvert(f3,v2.f3);
    FOR inx := 1 TO 10 DO
      v2.f4[inx] := f4[inx];
    hpFPConvert(f5,v2.f5,HP3000_32bit,IEEE_32bit,status,
              except,RoundToZero);
    SetConvert(f6,v2.f6);
    END; (*Assign the components*);
  Write(file2,v2);
  END; (*Read and Write*)
END. (*Program Convertfile*)
```

Exhibit 2

{This program manipulates real numbers as IEEE and calls a routine that will manipulate them as 3000 reals. It is responsible for all the real number conversions and uses the intrinsic, HPFPCONVERT to do them. Note that this routine is compiled without HP3000_16 because we want IEEE manipulation. It must be compiled separately from the routines that do 3000 manipulation.

The option $CHECK_ACTUAL_PARM 0$ is set to get rid of a bunch of linker warnings. Parameter checking needs to be turned off because the linker knows the difference between 3000 and IEEE reals and will generate a link error}

```
$CHECK_ACTUAL_PARM 0$
PROGRAM RealHPPA(Output);

{The procedure, RealAdd, adds two real numbers as IEEE and calls a routine
that adds them as 3000 reals and passes back the result. It converts the
result to IEEE real and prints it out. Reals are converted to 3000 real
before calling the 3000 add routine.}

PROCEDURE RealAdd;

CONST
  HP3000_32bit =1;              {Parameters for calls to HPFPCONVERT}
  IEEE_32bit = 3;
  RoundToZero = 1;

VAR
  r1_IEEE,r1_3000,
  r2_IEEE,r2_3000,
  r3,
  r4_IEEE,r4_3000 : real;
  status : integer;
  except: -32768 .. 32767;

PROCEDURE Real3000Add(
  r1,r2:real;
  VAR r3:real); EXTERNAL;

PROCEDURE hpFPConvert;INTRINSIC;

BEGIN
r1_IEEE := 1.3;
r2_IEEE := 1.2;
r3 := r1_IEEE + r2_IEEE; {done in IEEE}
writeln('IEEE value: ', r3);

{Convert reals to 3000 format for call the Real3000Add)

hpFPConvert(r1_IEEE,r1_3000,IEEE_32bit,HP3000_32bit, status,except,RoundToZero);
hpFPConvert(r2_IEEE,r2_3000,IEEE_32bit,HP3000_32bit, status,except,RoundToZero);

Real3000Add(r1_3000,r2_3000,r4_3000);
```

```
{Convert result back to IEEE. All reals in this routine are treated as
IEEE. So, conversion most be done before doing anything with the result.}

hpFPConvert(r4_3000,r4_IEEE,HP3000_32bit,IEEE_32bit, status,except,RoundToZero);
writeln('Converted back value: ',r4_IEEE);
END;

BEGIN
RealAdd;
END.
```

{This is the subprogram that does the 3000 real manipulation. It assumes that all the numbers it sees are in the 3000 format. $HP3000_16 is used to accomplish this. It applies to the entire compilation unit. Hence it must be compiled separately from part of the program that does IEEE reals. NOTE: All structures are HP3000_16 and hence are incompatible with all structures compiled without HP3000_16.}

```
$HP3000_16$
$SUBPROGRAM$
PROGRAM Real3000(Output);

PROCEDURE Real3000Add( r1, r2:real;
  VAR r3:real);

BEGIN
r3 := r1+r2;
writeln('3000 real value: ',r3);
END;

BEGIN
END.
```

HP3000_16 unit is in a source file called, real1. IEEE unit is in source file called real2. The commands are as follows:

```
:pasxl real1, real1obj
  .
  .
  .
END OF COMPILE

:pasxl real1,real1obj
  .
  .
  .
END OF COMPILE

:link from=real1obj,real2obj;to=realprog;parmcheck=0
INCOMPATIBILE PACKING: output (REAL1OBJ, REAL2OBJ)  (LINKWARN 1503)

:run realprog

IEEE value: 2.50000E+00
```

3000 realvalue: 2.50000E+00
Converted back value: 2.50000E+00

END OF PROGRAM

Note that the parmcheck=0 option to the linkeditor is necessary to prevent a type incompatibility error with output. $CHECK_ACTUAL_PARM 0$ is not necessary in the RealHPPA program because of this option. A lot more warnings would be given here instead.

Table of Contents

A Report on Report Writers

Stewart Hill
Hewlett-Packard Co.
Computer Language Lab
19447 Pruneridge Ave.
Cupertino, CA, 95014

INTRODUCTION

Reports have long been an integral part of business computing. For many years, report programs could only be written by trained programmers; such programs can be difficult to write and tedious to verify. But in recent years a growing number of languages and products have made report writing much easier. Many more people can now create and produce reports, even without training in standard programming languages.

Report Writers come in many different forms and with vastly different user interfaces. They range from statements imbedded in a programming language to forms-driven stand-alone report products. With the advent of friendly report writers, several questions must also be raised: 1) What kind of performance is achievable with a report writer? 2) What features should be expected from a report writer, and how might these vary from product to product? 3) Which users should reasonably have access to a report writer, and what users do not need such access?

This paper addresses the last two questions: what report writers do, and who should use them. Performance is addressed only in a general manner, as it is not within the scope of this paper to compare actual performance data. Such data has been obtained previously[1], and it is subject to change.

Several different report writers were examined in preparation for this paper, though these are by no means all that are available. The intent of this examination is not a product comparison, but an attempt to present the various implementation styles. This type of investigation brings out both the similarities and the differences which can be found in report writing systems. Additionally, these products are intended for different types of users, and looking at them side by side helps delineate the intended audience.

Each of these products stands on its own merits, and no judgements on them will be made. Figure 1 is a list of the products examined. These were chosen because reference documentation was readily available and because they represent the current range of user interfaces to report writers. The "style" listed is very general. The fourth generation languages do not all use the same style of interface.

---

[1] A Performance Comparison of HP3000 Report Writers, Roger W. Lawson, September 1987 INTEREX (Business Users) Proceedings

PRODUCT	COMPANY	Style
HP Business BASIC	HP	Imbedded 3GL
Powerhouse(r) QUIZ	COGNOS	4GL
HP Visor	HP	Forms-driven/SQL (HP-UX)
The WRITE STUFF	PROTOS	4GL
Business Report Writer	HP	4GL Report Specific
ASKPLUS	COGELOG	QUERY-based
COBOL Report Module	ANSI	Imbedded 3GL

Figure 1: Sample Report Writer Products

REPORT WRITER FUNCTIONALITY

What is a Report Writer?

To answer this question, one must first determine what is meant by a "report." In general, all output can be considered a report, no matter how unstructured it may be. However, such output is uncontrolled and far too general a definition to use. Instead, let us define a report to be the structured, formatted output of repetitive data, plus the ability to summarize that data. This definition certainly does not cover all reports, but it does describe the concepts behind most reports.

A report writer is a powerful controller. Its primary purpose is to control the structure of the output and the formatting of data within that structure. In addition, a report writer provides summary data for use within the report itself. The actual workings of a report writer are complicated and can be difficult to understand. Nevertheless, these controls relieve the programmer of an enormous amount of bookkeeping, which leads to better programs in less time.

To accomplish its goals, a report writer requires three distinct parts: a page layout controller, a data formatter, and a data retriever. In some products, these pieces are closely linked and are intended to be used only with one another. Other report writers provide these parts as independent controls which may be used with or without one another. Whatever the implementation, these controls are able to produce reports ranging from the very simple to the very complex.

Page Layout and Summary Controls

Report writers have two sets of controls for structuring report output. One set defines the layout of a page, which usually corresponds to a printed piece of paper. The other set controls the overall report, including the grouping and summarizing of data. All of the examined report writers provide these controls.

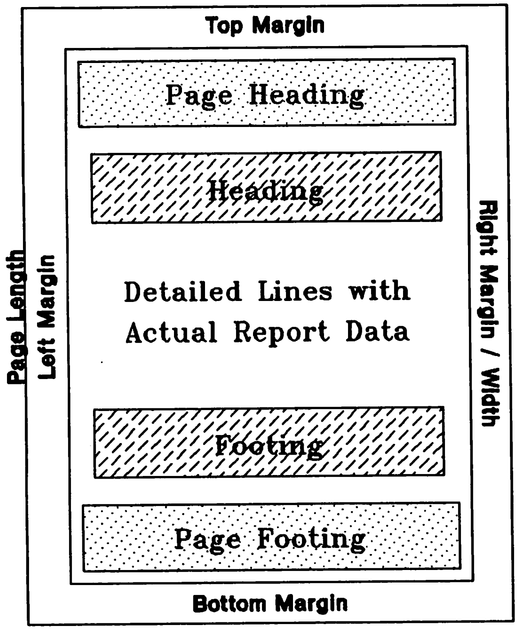


Figure 2: Page Layout

Figure 2 shows a typical report page, including some blocks associated with groups of data. All of the examined report writers define these parts of a page in some way. They also provide controls for customizing each area of the page. The terms used to describe the page layout are:

- **PAGE LENGTH**: The total number of lines on a page.
- **LEFT MARGIN**: The number of spaces printed before each line of output. Most report writers include this in the format of each line, rather than providing a specific control statement.
- **RIGHT MARGIN (or WIDTH)**: Indicates the last column in which data can be written. This is usually based on the output device.
- **TOP and BOTTOM MARGIN**: The number of blank lines which occur at the top and bottom of every page. Many report writers combine these with the PAGE HEADING and PAGE FOOTING controls, rather than having separate controls.
- **PAGE HEADING**: Defines data to be printed at the top of each page.
- **PAGE FOOTING**: Defines data to be printed at the bottom of each page.

- **HEADING:** Defines data to be printed before each summary grouping. (Described below)

- **FOOTING:** Defines data to be printed after each summary grouping. This area is often used to display totals or counts of the preceding detailed data.

Some additional controls for the layout are:

- **PAGENUM:** The current page number.

- **SET PAGENUM:** A means of modifying the current page number.

- **NEED:** A means of specifying that a certain number of lines must be on the page before the next output takes place. A new page is started if fewer lines are left on the current page.

NOTE

> The terms above do not represent syntax from any one report writer. These terms are commonly used and tend to convey their intent directly.

Among the report structure controls, the most important feature is the ability to define groups of data, called **summary groups.** These groups are usually based on a change in the value of a data base field or a variable. Such a value change causes a **break,** at which time the report writer may take some action. These breaks constitute the heart of report writer control, allowing great volumes of data to be printed and summarized in meaningful (and readable) reports.

Reports may use several breaks at the same time; these breaks are nested to provide different levels of detail with the report. The key to understanding report writers is understanding how these nested breaks work. This is relatively simple, but not necessarily straightforward. A careful reading of a manual, along with some hands-on experience, should be enough for most people to learn about a report writer.

The number of break levels (also called summary levels) available in the report writers varies, but nine levels appears to be the minimum number. Report writer controls allow a HEADING and a FOOTING to be defined at each break level, and summary data and automatic totalling can be kept for each level. These controls provide the programmer with great flexibility for designing the look of a report, and the ability to put out meaningful data.

The typical report structure controls are:

- **REPORT HEADING:** Defines data to be displayed only at the beginning of the report. Cover pages and introductions may be defined here. Not all report writers provide this facility.

- **REPORT FOOTING:** Defines data to be displayed only at the very end of the report. Often used to display grand totals for a report. All examined report writers provide this.

- **BREAK.** Defines the data base field, variable, or expression to use for grouping data. Multiple BREAK levels are allowed in order to define different levels of detail.

- **TOTAL**: Defines fields, variables, or expressions to be totalled automatically. This can be done at each break level. Usually a function is defined for printing this total, although some report writers do this automatically.

- **COUNT**: Provides an automatic count of the number of detailed items printed in a group. This value can be accessed at each break level.

- **AVERAGE**: Just returns TOTAL/COUNT. Almost all report writers provide this function.

- **REPORT**: Tells the report writer to process detailed data. This activates checks for BREAK controls, page boundary checks, and totalling. Some report writers provide this control automatically.

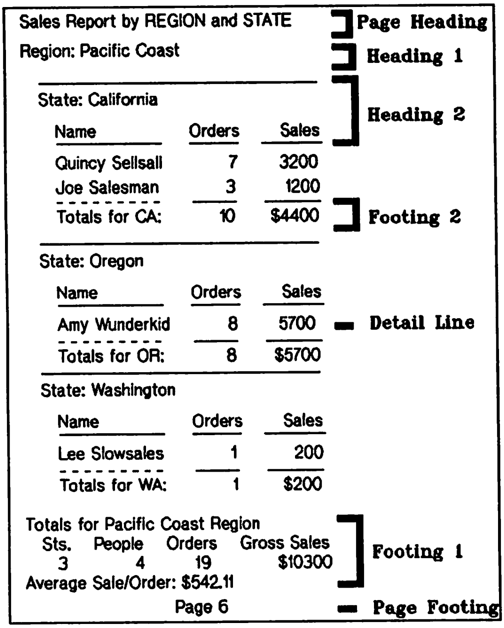


Figure 3: Report showing nested BREAK levels.

The example page shown in figure 3 is from a simple report which uses most of the report writer control features. In particular, two different break levels are shown, one for regions and one for states. The most detailed level is that for states; the employees and sales figures are shown at this level. The region level provides less detail: the only real information at this level is in **Heading 1** and **Footing 1**. There may be other break levels defined in this report as well. An example of this would be an even less detailed break, such as by country.

The program in figure 4 shows the control features that would be used to produce the example report page in Figure 3. Most of the actual printing has been left out, except to show the use of some control features. (The syntax used has been made up; however, the clarity of the syntax is typical of report writers which do not use visual interfaces.)

```
PAGE HEADING USES 2 LINES
      PRINT "Sales Report by REGION and STATE"
      BLANK                                     ! prints a blank line
PAGE FOOTING USES 1 LINE
      PRINT "Page " PAGENUM, CENTERED
HEADING 1 USES 2 LINES: BREAK IF Region CHANGES
      TOTAL FOR: Sales
      AVERAGE FOR: Sales
      COUNT OF: State, Orders, Employee
      ...
FOOTING 1 USES 5 LINES
      ...
      PRINT COUNT(State) COUNT(Employee) COUNT(Orders) Total(Sales)
      ...
      PRINT AVERAGE(Sales)
HEADING 2 USES 5 LINES: BREAK IF State CHANGES
      TOTAL FOR: Sales
      COUNT OF: Orders
      ...
FOOTING 2 USES 2 LINES
      ...
      PRINT "Totals for" State ":" COUNT(Orders) TOTAL(Sales)
...
! Now select the data from datasets and print the report
CHOOSE Region
      CHOOSE Employee, State, Orders, Sales
         SORT BY Region, State
         REPORT
```

Figure 4: Report Program

Additional Features

The controls listed above only define the minimal set needed for a good report writer. Most report writers provide some additional capabilities. Each new function only enhances the ability of a report writer to produce the desired output. One should always consider these features when attempting to choose a report writer. Fortunately, most capabilities beyond those listed can be found in several report writer products.

Typical examples of new capabilities are the addition of HIGH and LOW functions. These functions trace the highest and lowest value of break expressions. Such a function is not provided in all report writers directly, but the functions can be very useful. Another example is the addition of statements to suppress output. Such statements can be used to produce summary reports without defining a new report. Again, not all report writers have or need such statements.

Format Controls

Format controls provide the mechanisms for defining both line formats and item formats. Line formats control what each unique line of a report should look like, including the positions of data items in the line. Item formats control the display of each data item in a line, such as the format for numeric output. Since report requirements vary considerably, the formatting portion of report writers must be flexible and powerful.

Line Formatting

Line formats usually fall into one of two categories: "guide line" specifications and "columnar description" specifications. The type of specification depends upon the user interface for a particular report writer product. One could argue all day about which interface is better, but in reality both provide sufficient power for defining very complex reports. In any case, every unique line must be defined and then referenced by the report.

Guide Lines use a visual definition for each report. The user enters the line exactly the way it should be printed, using special markers for printing data in the line. Depending upon the product, this may occur directly in a file or report, or this may be done on a terminal. The report writer is then told when each line should be printed.

Columnar Descriptions combine data item formats with special functions to control the line format. The data item formats specify the field length for each item. Typical special functions are TAB, which skips to a particular column, and SPACE, which prints a specified number of blanks. The line format is built by indicating in what order data should be printed, with the output functions specifying the columns in which output should begin.

Some report writers provide a default output format. In these cases, the size and type of the data control the line format. In addition, the headings and footings may be given default formats as well. This is a convenient feature for accessing data quickly, but most reports will override these defaults to produce a better looking report.

Data Item Formatting

Virtually all report writers provide advanced formatting capabilities for individual fields. This allows the programmer to shape the report so that numeric and alphabetic data are placed properly in the report output. There are many ways to provide the formatting power needed, and each report writer provides its own specification technique.

Many report writer products use the COBOL PIC descriptor, or something very similar, to format data. This lends some familiarity to the report writer for many programmers and still provides a very flexible formatting system. Other report writers may mix COBOL formatting with their own formatting, or even provide a new type of format. Not much time is required to learn data format controls, so this should be of little concern to new users.

Data Retrieval

Reports are generally used to display large volumes of data, or to display small pieces of data from a large set. This implies that most reports will use data from a data base, rather than from files or from keyboard entry. Given this, it is understandable that most report writers are geared toward making data base access fast and simple.

All but one of the report writers examined run on MPE, and therefore use IMAGE or TurboIMAGE for data base retrieval. HP Visor runs on HP-UX, and it uses a relational data base. Syntactically, access to the data base differs greatly among these products. For example, COBOL and HP Business BASIC use intrinsics and built-in statements; HP Visor uses SQL or a form-driven access; the fourth generation languages all define their own statements for selecting data.

What became clear during this investigation is that the language type of the report writer is very important. That is, fourth generation languages (including HP Visor) provide easier access to the data base than third generation languages. Report programs in third generation languages are required to know much more about the data base definition than programs using the fourth generation languages. This generally means that 4GL report programs can be written faster and (arguably) more clearly than equivalent 3GL programs.

One fact is crystal clear. No one should have to write reports *without the aid of a report writer*. The bookkeeping combined with accessing the correct data makes for a very complicated task. Such programs are much harder to maintain and change than *any* program that uses a report writer.

Third Generation Language Access

There are two parts to data base structure: the definitions of and relationships between datasets (tables in a relational system), and the format of the data in each set. In a database like TurboIMAGE, the "format" includes the location of the data in a record as well as its size and type. This information must be made available to any programs which access the data base.

Standard languages usually access data bases through intrinsic or library calls. Some languages contain an interface to hide the details of these calls, but the net result is still a system call. The task of making these calls correct is left to the programmer. This implies that report programs assume or are told a great deal about the data base structure.

A report program may determine data base structure in many ways. Normally, the program source code tells which datasets to use, and how to link them together. Few programs verify the linking information at run-time. The data format may be obtained from a dictionary interface, or the programmer may code this directly. In any case, the programming language itself knows nothing about the data base, and the report program knows no more than the programmer gives it.

Third generation report programs are implicitly dependent upon the data base structure. The data base and the programs may change independently, and these changes are not automatically reflected in both. This may lead to errors when reports are finally produced. On the other hand, data base structures do not undergo radical changes often; such changes usually include time to update associated programs.

Fourth Generation Language Access

Fourth generation languages access data bases in a variety of ways. While most of the features turn out to be the same, syntax and performance vary greatly. Yet even more striking are the differences from third generation language programs.

As one should expect, fourth generation programs must specify what datasets (tables) to use, and which fields are needed by the report program. Unlike their third generation counterparts, however, these programs often attempt to define an automatic link between the datasets. In addition, these languages determine where the fields occur in the dataset, along with the size and type of each field. This field information is obtained without user intervention or specification.

The programmer's job becomes much easier once the report writer itself has access to the data base structure. Data fields may be used without regard to *exactly* how the data is retrieved. The process of linking multiple datasets becomes much simpler, even to the point of not being needed. The report program no longer needs explicit calls to retrieve data, as this is done automatically by the language statements. This allows the programmer to concentrate on the data required and the report format itself.

Most fourth generation languages provide easy data base access through the use of a dictionary or a schema definition. This interface may use a standard dictionary product or may require a separate definition file. Regardless, the languages can use this definition at run-time (or compile time) to provide the dataset linkage and the data formats. If minor (or even major) changes in the data base structure take place, the report programs take this into account automatically (or require a simple recompilation). Program source changes are required far less often.

Non-Data-Base Data Access

Data bases are certainly not the only source for report data, especially when the data is transmitted from an outside source. Data may reside on tape, in files, or be sent to the report program directly. In such cases, the data might be structured, but may also occur as "free-form" input.

Report writers provide a wide variety of access to alternate data sources. This ranges from accepting input from any source to restricting input to a data base. Third generation languages tend to be more adept at accepting any type of input; their I/O is normally disjoint from the report writer itself. Some fourth generation languages also do well, while others require a fixed format for each record processed. Still other products are based solely on access to a data base; in these cases, the data must be merged into a data base before the report can be generated.

Access to file and other source data is of little concern to most operations. The vast majority of reports are generated from data bases. This explains why most report writer products provide generous and easy access to data bases instead of concentrating on general input.

Performance Considerations

As with any program or language, performance is an important issue. Unfortunately, report writers are not always well understood, and consequently take the blame for poor performance. Report performance can be measured and in many cases improved. To be realistic, one must examine the three major pieces of report writers to determine where performance may suffer.

Report Writer Controls

The cost of the automatic report writer controls can be measured, albeit with some difficulty. To truly define their cost, however, a report program must be compared to an equivalent program which does *not* use a report writer. As stated earlier, writing such a program can be quite difficult.

Report writers pay a small penalty for providing generalized control statements. These statements can produce virtually any report, but this fact precludes many optimizations for specific reports. When a report writer is not used, a program may take advantage of knowledge about the incoming data or about the report layout. This allows a specific report program to minimize the data value checks and page checks which must take place during report output. On the other hand, very complicated reports may require just as many checks, and optimization may not be possible.

In general, report writer controls do not significantly decrease performance. Most of the work performed by these statements must be duplicated in hand-written report programs. The increased development and maintenance time for such programs outweigh the overhead costs for report writer controls.

Formatting

Performance improvements are indeed possible during data formatting, although the amount of improvement is very language dependent. Many report writer products give the user great latitude in controlling the output. More importantly, the improvements which might be made in a hand-written program are generally available in the report writers as well. In other words, the non-report-writer programs will not fare much better than a typical report writer program when it comes to data output.

The most likely way to improve report output speed is to change the report format itself. This allows faster data output specifications, thereby significantly reducing report output time. A typical situation occurs with currency signs in reports. The currency symbol can usually be "floated" so that it prints adjacent to a number; printing the symbol in a fixed location is much easier for the report writer. If currency symbols are printed on every line of a report, changing the format to use a fixed column (or even remove the currency symbol) will allow the report to be produced faster.

Data Retrieval

Data base access is by far the most critical performance aspect of any report writer. A report program may spend anywhere from *fifty to ninety percent* of its time in the data base! Obviously, any changes which can reduce this time will impact performance significantly.

Report writers do not generally prevent better data base access. Indeed, they should give the programmer more time to think about data base performance. Unfortunately, report writers sometimes make data retrieval too easy. The very powerful data base statements allow such easy retrieval that performance is overlooked. This fact becomes obvious when a report writer program replaces an old hand-written report program; the old program probably considers every aspect of performance, while the new program simply produces the same output without regard to performance.

There are some typical situations which occur during data base retrieval. One of the most common occurs when a program is based upon a request for a report, but the programmer does not closely consider the amount of data being retrieved. For example, suppose the following request is made:

```
Produce a report listing all orders over $10,000
which are 90 days or more past due.
```

This report is quite simple to produce, especially if the report writer has date arithmetic (many do). For example, the selection might be:

CHOOSE Order__amount >= 10000 AND DATE - Payment__due__date >= 90

There is nothing wrong with the selection criteria above. But suppose that the average order for this company is more than $10,000, and most of their customers pay on time. What performance can be expected in this case? The report writer will first find every order of more than $10,000, which will be most orders; then the check will be made for the number of days the bill is overdue. If this information resides in separate datasets, performance will be poor.

NOTE

> Report writers often implement *partial evaluation* for selecting data. This means that the evaluation of a second selection criteria depends upon the results of the first criteria. In the case above, the payment due date will not be checked unless the order amount is $10,000 or more.

In order to improve on this, the programmer must consider the likelihood of satisfying each condition; the report writer itself cannot do this. Performance can be increased greatly by changing the selection criteria:

CHOOSE DATE - Payment_due_date >= 90 AND Order_amount >= 10000

Now the order amount will only be retrieved if the payment is overdue. Since this does not apply to most orders, fewer data base records will be read.

Not all report writers allow absolutely optimal access to a data base. The more control a report writer gives to the programmer, the higher the chances of peak performance. But again, this must be traded off against the increased development and maintenance time for such a program. And the most important point is that programmers should not overlook performance just to get the report produced.

Third Generation vs. Fourth Generation Language

Historically, third generation languages yield report programs that run faster than fourth generation programs. While constant improvements are being made to fourth generation languages, one must consider this factor when deciding which report writer to use for a particular report. (Report generators which produce third generation programs should not be considered fourth generation products in this case.) In many cases, the performance gains here are not significant enough to consider changing from one language for another. These issues should be raised before the report program is started. But when performance does become critical, one must look at all the alternatives.

The friendliness and power of fourth generation languages comes from their ability to hide many implementation details from the user. That is, they let the user specify what to do without worrying too much about how things will be done. Unfortunately, this very aspect of the fourth generation languages means that they must typically do more work than an equivalent third generation language. Optimizations are more likely in third generation programs as well, since the programmer has direct access to the executing code. The net result is that third generation report programs run faster.

Many fourth generation languages interpret their programs, and they do this very quickly. Most third generation languages are compiled (some fourth generation products supply a compiler also). A compiled program can significantly outperform an interpreted program. For report programs, the performance difference depends upon how much time is spent in the data base; the difference is greater if less time is spent retrieving data. Again, these factors must be examined carefully before embarking upon a reimplementation of a report; the performance improvement might not be worth the time spent to rewrite the program.

MATCHING REPORTS, USERS, AND REPORT WRITERS

Clearly, many report writers provide the common functions needed to produce a report. But this does not mean that all report writers are equally easy to use, or that all types of users can make efficient use of any report writer. Different products are intended to be used by different users, and performance considerations may also influence the decision of what kind of report writer to use.

NOTE

The discussion that follows concerns the use of a report writer to create reports, not just to run a report. Only performance issues are relevant to the actual execution of a report.

There are several factors which influence the decision to use a particular type of report writer. Some of the important aspects of this decision are: the type of report, user background, user interface, and performance. The influence of each of these factors must be weighed when deciding what report writer should be used.

Users are not easily categorized, as their experience ranges from systems programmers to completely non-technical users. Proper report writer usage does depend upon the user's background; therefore, some groups must be defined. For this discussion, four types of users are defined: technical, semi-technical, occasional, and end users.

Types of Reports

Reports can be classified into a few fuzzy categories. These groups help determine the amount of work and the amount of experience needed to produce a report. A report may move from one group to another as modifications are made to the report, but the categories will still reflect the total amount of effort required. For descriptive purposes, these categories will be used: simple query-based reports, simple repetitive reports, complicated infrequent reports, and complicated repetitive reports.

Simple Reports

The query-based report represents the "on-the-fly" report, in which any data may be needed at any time. These reports are often run one time, or at irregular intervals. In many cases, a permanent report program cannot be written for such a report. Even when this is possible, this type of report may not be worth saving as a permanent part of a system. The query-based report is ideally suited to default report formatting, if the report writer provides this.

The simple repetitive type represents reports with straightforward page layouts and easy data access. These reports are run frequently. This type of report usually gets saved as a permanent feature to provide instant access to the report. Repetitive reports consist of known useful reports and query-based reports identified as being used frequently. Once again, default formats (if available) may be used with this type of report.

Complicated Reports

A "complicated" report refers to either a sophisticated report format, difficult data access, or, more often, to both. This report type implies that greater expertise, as well as a clear understanding of the desired output, are required. Complicated reports rarely use a default format when it is provided.

The frequency of producing a complicated report affects different factors in the report program. Infrequent reports allow the programmer to concentrate effort on the report format, with less emphasis on the performance of the report writer. **Repetitive** reports are run frequently or regularly. These reports tend to emphasize performance much more, especially when data base access is inefficient. These two types of reports require different tuning efforts to make them work well.

Types of User Interfaces

Report writers use three distinct user interfaces, which will be referred to as imbedded, programmatic, and visual. These interfaces are not mutually exclusive, and in fact several report writers provide facets of each, yet each interface seems to define its own set of users.

Imbedded report writers occur as statements in a standard, third generation language. These statements provide page controls and automatic totalling; the standard language provides the control over data base access and general output. The imbedded statements may extend output control or data base access as needed. The COBOL report module is an excellent example of an imbedded report writer.

The term **programmatic** refers to products which define their own language for the report writer. This includes fourth generation languages as well as some products specifically designed to produce reports. These report writers provide all controls necessary for writing a report, including data base access. Most programmatic products are reasonably small and friendly to use when compared to imbedded report writers.

Visual report writers use forms and other screen formats to guide the user. The forms allow a user to indicate what data to use, how to print the data, and what a page should look like. This type of interface is extremely easy to use, as no real "programming" is involved. Some report writers provide a visual mode as an alternative to the programmatic or imbedded interface.

Of course, reports may be generated by programs which do not use a report writer. This alternative is always open and must be considered when determining how a report program should be written.

The Occasional User

As the title implies, the occasional user does not need a report writer very often. This type of user relies on others to write report programs and possibly even run them. But every now and then, a report must be produced quickly, or privately, and the user cannot wait for someone else to write the program. The occasional user is *not* an end user; some technical knowledge is required. Managers are probably the best example of occasional users.

The infrequent and impromptu use of a report writer implies that query-based reports will be produced. Such reports are run in order to get immediate information. As a result, formatting and performance are not key areas of concern for most of these reports. If a report needs to be run repeatedly, even if only a few times a year, someone else can be assigned to write the report program.

The best interface for an occasional user is a visual one. This type of interface allows the user to see what will be produced before the report data is printed. In some cases, the report writer will guide the user until a minimal report can be printed. For simple query reports, this is ideal.

A programmatic interface provides a good environment for infrequent users. Data base access remains easy, and default formats are common in these products. The page layout and automatic summaries are still simple to use. However, programmatic interfaces tend to require more work on the user's part, as both statements and syntax must be remembered (or looked up). In addition, some knowledge of the data base is essential with this interface.

An imbedded language report writer is not designed for the occasional user. Too much knowledge of the language, the data base, and the report writer is required. Even a former technical user will have to look up a substantial amount of information. Very quickly the user will realize that the report takes too long to produce.

Most companies have relatively few occasional users. When choosing a report writer product, these users should not be a major factor in the decision process. The needs of the technical and semi-technical users far outweigh the needs of a few people to produce uncommon reports.

The Semi-technical User

Semi-technical users are a growing segment of the programmer population. This has been spurred by the tremendous growth of fourth generation languages, program generators, and user-friendly, specialized products. All of these products make programming much easier to understand, which has allowed many people to write custom programs.

The semi-technical user need not have any formal training or programming experience. Indeed, a user may not be able to write anything in a third generation language. This may prevent the writing of a general purpose program, but it does not prevent a great number of useful programs from being written. The semi-technical user is not a full-time programmer, but may have some programming duties. On the other hand, this user may use a report writer during spare moments just to improve the work environment.

This user group will create many more reports than the occasional user. Any simple reports can be produced, in addition to some complicated reports. The user's own expertise must be the guide to creating new reports; advanced users should be able to produce very fancy reports. Rarely used or custom complicated reports probably require too much time for the semi-technical user to create; repetitive reports also require time, but often are worth the person's effort.

Both visual and programmatic report writers are available and used by these users. One could debate for days about which style is better. The important point is that the report writer provide the capabilities needed by the semi-technical user. As a person writes more reports, the sophistication of the reports can increase. The report writer should not impede this progress.

Imbedded report writers may be used by some semi-technical users, but in general this is not the case. General purpose languages are more difficult to learn and to use than fourth generation products. Even the size of a standard language can be daunting for some users. In addition, the relatively unfriendly environment for writing programs will drive away most semi-technical users.

Unlike reports produced by the occasional user, the performance of a semi-technical user's report program can be important. As indicated earlier, a small change in data base access can significantly impact system performance. Many users are not trained to recognize this situation, and even fewer know how to make effective changes. One must rely on the more advanced users or a "technical" user to help streamline these report programs. The choice of a report writer which allows this streamlining is important if many semi-technical users are writing programs.

The Technical User

The technical user category represents full-time programmers. These people often have formal training or much experience in the computing field. A background in third generation languages is very common, but not strictly necessary for report writers. Understanding how a program or system works is much more important than language knowledge. The technical user employs this knowledge to write efficient report programs.

The technical user should be called upon to write the most complicated reports. Such reports can get bogged down in very exacting details which usually frustrate the more casual users. In addition, technical users should write or review any repetitive report programs, especially when they are run frequently. This ensures that the report programs attain the best performance.

All styles of report writers are usable by technical users, including languages which do not support report writer statements. Experience has shown that these users can produce fairly sophisticated reports very quickly when using either visual or programmatic report writers. Imbedded report writers require more time to use, but not nearly as much as a language with no report writer. The experienced user will weigh several factors before choosing what language to use for a report.

It is worth noting that not all report writers are equal for a technical user. There are reports which are extremely difficult to write in fourth generation languages. There are also reports which require great control over data base access in order to perform quickly; not all report writers provide the necessary control. On the other hand, *most* reports use very straightforward data base access and do not need such tight controls.

Performance is a key issue for technical users. The programmer must help the semi-technical user understand why a report runs slowly, and must help that user improve the report program. To do this, a thorough understanding of the report writer is needed, particularly with respect to data base access. The technical user's own report programs can take performance into account immediately.

Is one report writer enough?

Undoubtedly, many people would answer this question with a resounding "YES!" For occasional and semi-technical users, this answer is certainly correct. But a technical user has many factors to consider, and it may be that the correct answer to this question is actually "NO." Some of the key factors in this decision are performance, integration, and responsiveness.

Performance should be obvious by now. Every report writer creates programs with different performance. If performance is critical, an imbedded report writer may work best. On the other hand, many programmatic report writers provide sufficient controls to yield near maximal performance.

Integration is a tricky issue. This area refers to the need to merge the report program into an overall system. Some report writers are part of a much larger system, and so integration becomes an easy topic. But there are other report writers which run alone; the report programs from these products may have to be merged with other pieces to build a system. The technical user must determine the feasibility of this merger, as well as its affect on the user's system interface.

Responsiveness refers to the speed with which a new report can be created. If a request for a new report comes in, the technical user must determine how soon the report has to be available. Fourth generation products allow much faster response, in general, but this may be outweighed by other factors. This decision must be based on the business environment and customer needs.

Of course, the technical user may already have a report writer. In this case, the factors above, along with other possible motivations, may indicate that a second report writer is needed. The second report writer should not be the same style of program. For example, owning two programmatic report writers is probably not useful, but owning a programmatic and an imbedded report writer might be. This decision should not be made lightly, but it is worth considering.

THE END USER

Finally we come to the end user. At some point, everyone is an end user; all one has to do is run a program or an application. But the end user under discussion is the true end user. This person has little or no computing background and does not have the need to learn. Programming is not within the scope of this user's job. The user only wants to get information from the computer.

Given the user's background, a valid question to ask is "Should the end user be able to create new reports?" The answer should be no. Creating a report requires knowledge in two technical areas: 1) the report writer language or interface, and 2) the data base description. Some report writers are extremely simple to use, but one must still understand the terms used. Many users do not or cannot take the time to learn these details. In many cases, end users have neither the need nor the time to learn about a report writer.

Data bases can be even more confusing. End users with no computing background have no concept of how data bases are arranged, or why they are defined in a particular manner. Data base field names are often abbreviated, and there are fields which users know nothing about. In addition, security may dictate that end users remain unaware of some data base information. This precludes publishing the data base format for the end user, leaving no way to access the necessary information.

Some end users may have a genuine need to produce query-based reports. These users should be properly trained to use a report writer. The training itself moves the user into the semi-technical class. This will not change the user's job responsibilities; it will give the person the capability to do the job properly. Not all end users need to be trained; a few people can be trained and all query-based reports can be redirected to these users.

Assuming that the end user cannot be trained, what would it take to allow end users to produce query-based reports? Simply put, this requires a much more sophisticated interface than report writers provide. The interface must be customized to the job at hand, which indicates that several interfaces may be needed. The reports must be very easy to produce and the data base must be hidden completely. The capability to write an end user interface does exist today, but the programs can be quite complicated. A better alternative might be to write small report programs for the most common queries, and pass the less frequent queries on to someone with more training.

A Comparison of TurboIMAGE and HPSQL

by Larry Kemp, HP Bellevue, WA

This paper is intended as a primer on HPSQL for current users of the IMAGE database management system on HP3000 computers. SQL, which is an acronym for Structured Query Language, is the new relational database management system for the HP3000 family. SQL was originally implemented on IBM mainframes, and has since been implemented on several other computer systems. SQL is an implementation of the original "System R" specification for relational databases. The ANSI committee has accepted SQL as the relational database model.

Users of TurboIMAGE will find that HPSQL provides considerably more flexibility than does TurboIMAGE. IMAGE has probably gained most of its popularity due to its ease-of-use and simplicity of design aspects. SQL should provide even more ease-of-use and simplicity.

IMAGE has gained popularity due to its good performance, predominantly to do with the ease with which the designer can take performance into account. For example, the IMAGE designer can effectively, easily, and accurately utilize blocking factors.

Another area where IMAGE excels is having a considerable knowledge and experience base. IMAGE is installed on all HP3000 computer systems, and IMAGE is the database management system used for most HP3000 applications. Therefore, there is considerable expertise available on good IMAGE design, both from HP and from a large number of third party consultants. The IMAGE handbook exemplifies the public knowledge base. There are a number of well known implementation (and optimization) techniques for IMAGE.

There is a knowledge base for SQL, and for the most part that knowledge focuses on high level design issues. There are well documented logical database design techniques that utilize relational database constructs, one example which is the normalization of databases to "third normal form".

The last, very positive trait of IMAGE has been its reliability. IMAGE databases rarely, if ever have integrity problems. And when some damage does happen, there are accurate, if not time consuming, recovery techniques. Since SQL is new, its reliability remains to be seen. SQL does have automated logging and rollback recovery, so SQL databases should not have integrity problems.

The remainder of this primer will focus on the usage and features of IMAGE and SQL on a sample database and problem. I will focus on data structure and design, query (data manipulation) language, program-and-data independence, security, and transaction management. I feel that these are the reasonings for databases.

Structure.

IMAGE and HPSQL use different terms to describe database structure. IMAGE uses the term "sets" to describe logical groupings of like described data. A non-database user would call that construct a file, with a restriction that all of the records are of the same record-layout. An SQL user calls that construct a "table". The IMAGE user refers to repetitive occurances in the set as "entries", while the non-database user refers to that construct as records.

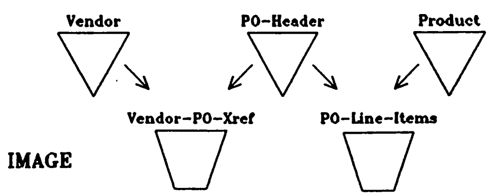
The SQL user refers to those constructs as "rows". And lastly, the IMAGE user refers to the individual components of an entry as "data items", where the non-database user refers to them as fields. The SQL user refers to "columns".

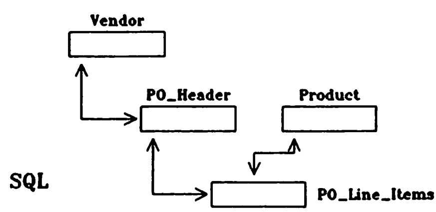
Non-Database	IMAGE	SQL
File	Set	Table
Record	Entry	Row
Field	Item	Column

IMAGE datasets are defined as one of master datasets, or detail datasets. Master datasets have unique keys and can be accessed by key or sequentially. Entries in a detail dataset are chronologically organized by common key. Entries can be accessed either sequentially, or along the chronological key path. Master datasets can be related to details, and in a logical sense, detail datasets can be related to masters. This results in the definition of IMAGE has an extended two level hierarchy.

SQL makes no distinction of master versus detail datasets. Any two tables can be related, allowing multi-level "Join" operations. And any table can be accessed either by key/path, or sequentially. Furthermore, a given table can have multiple keys, including keys which are formulated from several columns. Generic and approximate searches are allowed.

Here is an example implementation using the two database management systems:





The most noticeable difference between the two implementations is the lack of an artifical connecting dataset between Vendor and PO-header in the SQL database. Just as worthy, is that the SQL database implements the same functionality as the IMAGE version. SQL, as in IMAGE, has the ability to declare unique keys for both the Vendor and PO-header tables. Also, the PO-number index for the PO-line-items dataset can be declared "clustering", which allows optimized physical placement along that index, in an analogous technique to the "primary path" for IMAGE detail datasets.

Query Language.

Most users of IMAGE were introduced to IMAGE through QUERY. QUERY has a simple, English-like syntax that allows command driven access to IMAGE databases. Query is a good learning tool in that it is easy to learn, and allows exercising of most IMAGE functions. Once the novice has mastered QUERY, he/she next learns how to programmatically access IMAGE. This involves formatting subroutine calls to IMAGE. One record is accessed at a time, with one or two calls necessary to access each record.

SQL, like IMAGE, has an ad-hoc program for accessing databases. (The SQL program is called ISQL, where the I stands for Interactive.) Most SQL users learn SQL through ISQL, in an analagous manner to the IMAGE user with QUERY. But unlike the IMAGE, programmatic access to SQL is nearly identical to ISQL access. In other words, the user codes the same commands programmatically as he/she uses in ISQL. Consequently SQL is easy to learn.

Like Query, SQL allows conditional specifications of rows to be selected. And like Query, SQL uses that specification to determine the access method. The access method is determined by SQL, and not by the application program.

SQL implements the query language in COBOL and PASCAL by using pre-processors. These pre-processors translate the high-level query commands into the appropriate subroutine calls. The only difference between the interactive commands and the programmatic SQL commands are the specification for where the resulting data resides. (Programmatically, the INTO clause is specified which says where in the program to store the result of a command.)

Here are sample interactive and programmatic SQL commands:

```
SELECT NAME FROM VENDOR                WHERE VENDOR_NUMBER = '0023'

SELECT NAME FROM VENDOR INTO :WS-NAME WHERE VENDOR-NUMBER = :WS-VENDOR-NUM
```

In this example, NAME is a column in the table VENDOR. A row with the VENDOR-NUMBER equal to the value of WS-VENDOR-NUM in the COBOL program is selected. And from the selected row, NAME is delivered to the COBOL item WS-NAME. Notice that the only significant difference is the specification of program data names in the programmatic version. (You might also notice the substitution of '-' for '_'. SQL syntax wants an underscore, while COBOL wants dashes, so the preprocessor converts dashes to underscores.)

Additionally, SQL commands have the ability to retrieve multiple records in a single command. This eliminates the need to code loops in many transaction processing programs. (It also speeds up performance, since it reduces the number of entries and exits from SQL.)

Using the sample database, here are code comparisons for IMAGE versus SQL. These statements display a purchase order. The SQL code is actual code, where the IMAGE code is psuedo-code. Notice that the SQL version takes exactly one SQL command to retrieve all qualifying rows.

```
SQL:  BULK SELECT * INTO :PO-RECORDS
        FROM PO-HEADER,VENDOR,PO-LINE-ITEMS,PRODUCT
        WHERE PO-HEADER.VENDOR-NUMBER=VENDOR.VENDOR-NUMBER AND
              PO-HEADER.PO-NUMBER=PO-LINE-ITEMS.PO-NUMBER AND
              PO-LINE-ITEMS.PROD-NO=PRODUCT.PROD-NO AND
              PO-HEADER.PO-NUMBER = :WS-PO-NUMBER

IMAGE:    DBGET(MODE7,PO-HEADER,PO-NUMBER,PO-RECORD)
          DBGET(MODE7,VENDOR,VENDOR-NUMBER,VENDOR-RECORD)
          DBFIND(PO-LINE-ITEMS,PO-NUMBER)
          REPEAT
            DBGET(MODE5,PO-LINE-ITEMS,LINE(I))
            DBGET(MODE7,PRODUCT,PROD-NO,PROD(I))
            ADD 1 TO I
          UNTIL (END-OF-CHAIN(PO-LINE-ITEMS))
```

This particular example is a complex one, requiring accesses to four different data sets or tables, and locating multiple records from the PO-LINE-ITEMS dataset or table. Note that the SQL user can test out his/her query interactively, using ISQL, before coding the command.

Program and Data Independence.

One of the most significant advantages of a database system is the ability to change the database without affecting the executing programs. All database systems have this characteristic to some extent, really none completely implement it. (One example is where a program accesses a field that has been eliminated from a database.)

IMAGE allows addition and deletion of fields of a database by a database administrator. IMAGE allows changing of field definitions by the database administrator such that programs that do not access the changed fields need no modifications.

The mechanism that IMAGE uses to implement this feature is called access by "item list". Specifically, when a program asks IMAGE for data, it presents a buffer, and a symbolic list of data items that describe the items that should fill the buffer. For example, a program might present IMAGE with the item list "VENDOR-NUMBER,VENDOR-NAME".

IMAGE databases are at least initially created by a text file called a "schema". A database administrator creates the schema which defines all sets, items, relationships, and security. Subsequent structural changes can be made to the database by modifying the schema, and recreating the database, or by use of Adager, or a similar utility which recreates the affected datasets. In all cases, the changes are made offline, and the database administrator will probably want to maintain the schema file.

Like IMAGE, SQL provides item flexibility by having programs request data using an item list. SQL also provides a significantly greater degree of program and data independence through a construct call a "View". A VIEW is a logical window that a program uses to access the database. A VIEW might be construed as a logical 'table' in that a program accesses a view just as it might access a table. A view can contain join operations across multiple files.

Here is an example of a VIEW:

VIEW creation:

```
CREATE VIEW PURCHASE_ORDER (PO_NUMBER,VENDOR,AMOUNT) AS
  SELECT PO_HEADER.PO_NUMBER,VENDOR_NAME,AMOUNT
    FROM PO_HEADER,VENDOR
    WHERE PO_HEADER.VENDOR_NUMBER=VENDOR.VENDOR_NUMBER
```

VIEW access (which could be programmatic):

```
SELECT * FROM PURCHASE_ORDER WHERE PO_NUMBER = '1020'
```

The VIEW facility allows external specification of not only the data elements accessed by a program, but also the access path to the data. It allows a program to retrieve data with no knowledge of the access path. It also allows the access path to be changed without requiring alterations to the program.

SQL databases are maintained by SQL commands. These can be given interactively or programmatically, just as any other command. Physical database structure changes can be made while the database is in use. For example, the following command could be given while the specified table is in use:

```
ALTER TABLE VENDOR ADD CLASSIFICATION CHAR(2)
```

This command would add a new column CLASSIFICATION to the table VENDOR. Currently executing programs would not be affected.

The following command could also be given while the database is in use:

```
CREATE INDEX PO_LINE_ITEM ON PO_LINE_ITEMS(PO_NUMBER,PART_NUMBER)
```

This command creates a combined index for the table PO_LINE_ITEMS using the columns PO_NUMBER and PART_NUMBER. Applications using the original database and selecting on PO_NUMBER and PART_NUMBER would have used the PO_NUMBER index, and then searched sequentially for the PART_NUMBER. Now those applications can use the new index PO_LINE_ITEM to go directly to requested line item. This change in access method is transparent to application programs.

Security.

There is little doubt in the industry today that security is an important job of a database management system. Ad-hoc programs, third-party applications, and open computer systems have mandated externally managed security systems.

IMAGE implements security in the form of passwords. Data items and data sets are passworded for a combination of read/update/none access to data items and read/write/none access to data sets. Passwords are specfied by the application program when it opens the database.

Security in SQL is implemented through the granting of access rights to logon user ids. Rather than use a separate password, SQL uses the user logon id, and allows MPE security to be used for passwording. Access is granted against tables or views. Since access to elements can be restricted by using views, data element security is achieved. Hopefully, this will prove to be a simpler technique.

A view, however, is more than simply a subset of data. It can contain not only access specification, but also selection criteria. Since access to data can be granted on views, this allows security to be specified by value. For example:

```
CREATE VIEW PO23 AS
   SELECT * FROM PO_HEADER WHERE VENDOR_NUMBER = '0023';
GRANT SELECT ON PO23 TO VENDOR23@PURCH;
```

This view allows the user VENDOR23 in the account PURCH to look at only his own purchase orders in the PO_HEADER table.

Transaction Management.

One of the functions of a database management system is to coordinate data between concurrent users. There are two issues: (1) protection against "race conditions" where multiple users desire to access and update the same data, and (2) guaranteeing logical integrity of data. A database management system protects against race conditions by serializing access to the same data. And a database management system guarantees logical integrity by ensuring that either all of its database manipulations succeed, or none of it succeeds.

For example, a user is going to make a transaction which adds one part to inventory, and subtracts one part from a purchase order. The increment to inventory includes reading the data and then updating it. No other updating transaction can be allowed to intervene between the read and update. If an intervening transaction did update the inventory count, then this transaction would make its changes to inventory using the old inventory count, effectively undoing the other transactions inventory update. In the case of system or program failure, the transaction must either have completed, or must be backed out. Otherwise, the partially completed transaction might allow artificial inventory growth.

IMAGE has two facilities to address transaction management: Locking and Transaction Logging. Locking allows programs to logically reserve a specified item before making a transaction against it. Locking is done explicitly by the program. Transaction logging allows a program to declare the beginning and ending of a logical transaction. In the case of system failure, the database can be recovered to last consistent (logically complete) point before the failure. Here is an example of inventory receivings using the sample database:

```
DBLOCK(product.prod-no=2666,po-line-items.prod-no=2666)
  DBGET(product,prod-no=2666)
  REPEAT
     DBGET(po-line-items,prod-no=2666)
  UNTIL (po-number=A2345)
  DBBEGIN
    DBUPDATE(product,qty-on-hand)
    DBUPDATE(po-line-items,qty-received)
  DBEND
DBUNLOCK
```

In this example, the program locks the item PROD-NO in both the PRODUCT and PO-LINE-ITEMS datasets. Then it retrieves the requisite entries. Once the entries are found, then a logical transaction is started which updates quantity fields in both datasets.

SQL implements the same constructs, but using a more automated technique. Locks in SQL are implicit; the programmer never needs to code LOCKs into a program. SQL determines concurrency conflicts by examining the data "pages" (which are similar to blocks) accessed within a transaction. If one transaction conflicts with another transaction, then SQL will either wait for the other transaction to complete, or return an error, allowing the transaction to restart itself.

This technique is not only eaiser to use, but it can also be more efficient. For example, a transaction which updates a bill-of-materials has no idea at the start of the transaction which part-numbers to lock, since the parts-explosion is determined by reading the records to be updated. An explicit locking technique would require either data set locking, or double accesses to the parts dataset. The SQL technique allows maximum concurrency since it does not require pre-determined locking.

Here is the SQL version of the parts receiving problem:

```
BEGIN WORK;
  UPDATE PRODUCT SET qty_on_hand = qty_on_hand + 1
    WHERE prod_no = '2666';
  UPDATE PO_LINE_ITEMS SET qty_received = qty_received + 1
    WHERE prod_no = '2666' AND po_number = 'A2345';
COMMIT WORK;
```

SQL assures logical consistency of data using a similar technique to IMAGE: each transaction is bracketted by the commands BEGIN WORK and COMMIT WORK. After a system failure, the uncompleted transactions are backed out in a manner similar to IMAGE. SQL provides the additional feature that if a program aborts, that any incomplete transactions will be rolled back.

A transaction roll-back can also be programmatically initiated by the ROLLBACK WORK command. This feature can simplfy, and potentially optimize transaction processing programs. For example: a program to fill sales orders might first match requested line items against inventory to see if the order could be filled. If the order can be satisfied, then it would re-read, and update the records from inventory. The SQL version of this program would simply read and update inventory. If a line item could not be satisfied from inventory, then it would request a rollback.

In Summary.

SQL provides all of the features of IMAGE, and in most cases in a significantly enhanced fashion. Additionally, SQL allows the user to administer databases at a considerably higher level. With SQL, the database administrator has a high degree of program independant control over data and access paths. In essence, SQL has provided the database adminstrator with many of the tasks that require programming (and debugging) in IMAGE.

The simplicity of IMAGE has resulted in very good performance for well designed IMAGE databases. IMAGE performance is well understood and reasonably consistent.

On the positive side of performance for SQL is that when a database performance issue arises after an application has already been implemented, that the database administrator can take action without involving changes to program logic. In other words, SQL allows the database designer to make mistakes in the initial design, and correct them after the fact.

References.

Astraham, M. M., et al, "System R: A Relational Approach to Database Management," *ACM Transactions on Database Systems 1*, No. 2 (June 1976).

Codd, E. F., "Normalized Data Base Structure: A Brief Tutorial", Proc. 1971 ACM SIGFIDET Workshop on Data Description, Access and Control. Available from ACM.

Codd, E. F., "Further Normalization of the Data Base Relational Model", in *Data Base Systems*, Courant Computer Science Symposia Series, Vol 6, Prentice Hall (1972).

Russell, Marguerite (ed.), *The IMAGE Handbook*, Wordware(1984), Seattle, WA.

Also, TurboIMAGE and HPSQL Reference Manuals from Hewlett-Packard Co.

Comparison of Limits.

	TurboImage	HPSQL
Sets/Tables per Database	199	Unlimited
Items/Columns per Database	199	Unlimited
Items/Columns per Dataset/Table	255	64
Item Size	4094	3996
Items/Columns per Path/Index	1	15

Sample SQL/COBOL program.

This program prompts the user for a product number, and displays all purchase orders against that product, including which vendor that the purchase order was issued to. Each SQL statement is bracketted by EXEC SQL/END-EXEC, to signal to the pre-processor that these are SQL commands. Notice that the pre-processor also knows the data-division elements, allowing it to check on data types and lengths.

```
IDENTIFICATION DIVISION.
PROGRAM-ID. POS.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
01  I PIC S9(4) COMP.
EXEC SQL INCLUDE SQLCA END-EXEC.              <<SQL communication area>>
EXEC SQL BEGIN DECLARE SECTION END-EXEC.      <<SQL data elements>>
01  PURCHASE-ORDERS.
    05  PURCHASE-ORDER OCCURS 20 TIMES.
        10  PO-NUMBER PIC X(6).
        10  VENDOR-NAME PIC X(20).
        10  QUANTITY PIC S9(4) COMP.
01  PROD-NO PIC X(4).
01  ERROR-MSG PIC X(72).
EXEC SQL END DECLARE SECTION END-EXEC.
PROCEDURE DIVISION.
OPEN-DATABASE.
    EXEC SQL WHENEVER SQLERROR GO TO SQL-ERROR END-EXEC.
    EXEC SQL CONNECT TO 'PURCHDB' END-EXEC.
ASK-FOR-PART-NO.
    DISPLAY 'ENTER PROD-NO FOR INQUIRY, OR CR TO STOP'.
    MOVE SPACES TO PROD-NO.
    ACCEPT PROD-NO.
    IF PROD-NO EQUAL SPACES THEN GO TO CLOSE-DATABASE.
    EXEC SQL BULK SELECT PO-LINE-ITEMS.PO-NUMBER,VENDOR-NAME,QUANTITY
             INTO :PURCHASE-ORDER
             FROM PO-LINE-ITEMS,VENDOR
             WHERE PO-LINE-ITEMS.PO-NUMBER=PO-HEADER.PO-NUMBER AND
                   PO-HEADER.VENDOR-NUMBER=VENDOR.VENDOR-NUMBER AND
                   PO-LINE-ITEMS.PROD-NO = :PROD-NO END-EXEC.
    IF SQLCODE GREATER THAN 0 THEN DISPLAY "NO POS FOR THAT PART-NO"
    ELSE PERFORM DISPLAY-PO VARYING I FROM 1 BY 1 UNTIL I > SQLERRD (3).
    GO TO ASK-FOR-PART-NO.
DISPLAY-PO.
    DISPLAY "PO-NUMBER=" PO-NUMBER (I)
        "   VENDOR-NAME=" VENDOR-NAME (I)
        "   QUANTITY=" QUANTITY (I).
SQL-ERROR.
    EXEC SQL SQLEXPLAIN :ERROR-MSG END-EXEC.
    DISPLAY ERROR-MSG.
CLOSE-DATABASE.
    EXEC SQL RELEASE END-EXEC.
    STOP RUN.
```

The Advance of HP SQL - Towards the OLTP Market

Alberto Lutgardo

Distributed Data Management Lab
Hewlett-Packard
19447 Pruneridge Avenue
Cupertino, CA 95014

Abstract

One of the attributes of an Online Transaction Processing system is to provide to the user a high level of throughput. The purpose of this paper is to explain to the user how to increase the throughput of a given application by the use of the Isolation Level concept, and how this concept was implemented in the HP SQL product. This paper shows the relationship between the concept of Isolation Level and the concept of consistency levels.

Isolation Level is a property of data base access from the point of view of concurrency control while the level of consistency is defined as the effects the user experiences due to the concurrency control. There are three Isolation Levels that stand among all. The Repeatable Read is the HP SQL default. It implies that all locks acquired by a transaction are held until transaction commits, rolls back, or it is terminated by the system due to deadlock. Cursor Stability holds the read locks only until after a subsequent read occurs. Read Uncommitted does not acquire locks for read operations; in other words, it only provides physical consistency. Write locks are the same for all isolation levels and are held until transaction commits or rolls back.

1. Introduction

HP SQL is a transaction oriented system; therefore, it will provide a high level of transaction throughput. This is vital for a product to be competitive in the OLTP market. An increase of concurrent access to common data increases the transaction throughput in the system. There are two ways to increase concurrency to common data. The first is to relax the locking scheme by using weaker locks, and the other one is to release the locks on common data as soon as possible. The HP SQL product has implemented the concept of Isolation Level to achieve this purpose. The HP SQL product provides the option of Isolation Levels to allow the user to control the concurrency of his/her applications.

This paper assumes that the reader is familiar with the concepts of concurrency control as defined in [2] and also familiar with HP SQL concurrency control mechanism as described in [5].

The concept of Isolation Level in the HP SQL product applies only to PUBLIC tables. A PUBLIC TABLE is a table that allows multiple writers and multiple readers concurrently. Cursor Stability is one of the isolation levels implemented in the HP SQL product that releases read locks before the transaction is committed. Cursor Stability releases read locks between SQL FETCH commands. The feature is provided for applications that scan a table in one direction.

The organization of the rest of the paper is as follow: Section 2 provides the HP SQL definition for Isolation Level. Section 3 describes the implementation of Isolation Level within the HP SQL product. Section 4 describes an application that takes advantage of the Isolation Level concept to increase the throughput of the system. Section 5 describes some performance results. Section 6 offers some conclusions, and it describes new features that HP SQL is planning to provide to the user in the near future to be even more competitive in the OLTP environment.

2. Isolation Level Definition

The concept of Isolation Level has been reported as a level of consistency in [2,3,4]. Level of consistency is defined as the kind of anomalies that can occur in the system during the execution of concurrent transactions. Some of these anomalies are:

- "dirty read": a transaction T1 can read a tuple that has been just created by another transaction T2 which is aborted after T1 has read the tuple.

- "non-repeatable read": transaction T1 is not guaranteed to get the same tuple if it re-reads it.

- "phantom read" transaction T1 is not guaranteed that it will get the same set of tuples matching a given search condition if it re-reads the table using the same search condition

The level of consistency for 0, 2, and 4 are defined in [4]. The user might experience all three anomalies when he uses consistency level 0, with consistency level 2 the user might experience non-repeatable reads and phantom reads, and with consistency level 4 the user will not experience any anomaly.

HP SQL defines Isolation Level as a property of an access to a table, defined in the data base, through a cursor from the concurrency point of view. A cursor in the HP SQL is an address of some specific tuple in a table. A cursor is associated with a type of access to a table. The HP SQL product provides internally index access and sequential access. A cursor can be defined explicitly by the user through the SQL DECLARE command or implicitly by the HP SQL product. There are eight isolation levels of which The HP SQL product currently support five. The relationship between Isolation Level and level of consistency is shown in Figure 1.

A description of each Isolation Level and an example of how to use it using the "accounts" PUBLIC table are provided below. Let us assume that the "accounts" table has four columns: account number, account name, debit, and credit. Each example shows the type of locks that the HP SQL product acquires on behalf of the transaction. However, in order to describe the concurrency between transactions, two concepts must be introduced first. READ ONLY TRANSACTION is an HP SQL feature under investigation in which updates are not allowed, while in a READ-WRITE TRANSACTION, retrievals and update are allowed. Only the READ-WRITE transaction is currently supported by the HP SQL product.

The definition of each Isolation Level is based on the set of tuples that the transaction retrieves through a cursor and the protection that this transaction provides to the set from the concurrency point of view. A cursor is associated with the type of scan done to a table. There two types of scans, index scan and relation scan.

Isolation Level	Level of Consistency
Repeatable Read with Intent Update	4
Repeatable Read	
Phantom Read with Intent Update	3
Phantom Read	
Cursor Stability with Intent Update	2
Cursor Stability	
Read Committed	1
Read Uncommitted	0

Figure 1

2.1 Repeatable Read

The set of tuples obtained through a cursor, matching the search condition, within a transaction can not be updated by another transaction; however, it can be read by another READ-WRITE transaction. No data can be added to the table that matches the search condition of the cursor through another transaction. Therefore, the same set of tuples are generated if the data is re-read using the same cursor.

```
EXEC SQL BEGIN WORK RR;

EXEC SQL SELECT * FROM accounts WHERE
                    credit > 1500.00;

EXEC SQL UPDATE accounts SET credit = credit*1.1
                    WHERE debit = 0;

EXEC SQL COMMIT WORK;
```

In order to explain the type of locks that the HP SQL product acquires on behalf of the SQL SELECT command, we can loosely say that the SQL SELECT command is divided internally into a) open scan, b) several internal FETCH commands, and c) close scan.

- Index scan: Subshare lock on the "accounts" table is acquired by the HP SQL product at open time. When the internal FETCH command is executed, a share lock is acquired by the HP SQL product for index and data pages. Once the SQL UPDATE command is executed, the subshare lock in the "accounts" table is promoted from subshare to subexclusive. Besides, for each updated page, there is a promotion of the lock from share to exclusive.

- Relation scan: Share lock is acquired by the HP SQL product on the "accounts" table at open time. When the internal FETCH command is executed, no locks are acquired by HP SQL product. Once the SQL UPDATE command is executed, the share lock on the "accounts" table is promoted to share subexclusive.

- Concurrency: Other READ/WRITE transactions can be executed concurrently with the above transaction between the SQL SELECT and the SQL UPDATE commands. Deadlocks can occur in this scheme during lock promotion.

- Usability: This Isolation Level should be used if the set of tuples to be retrieved will *not* be updated within the transaction. The user should either specify or let it default to the RR option in the SQL BEGIN WORK command. If the user only wants to retrieve and not update the data using the SQL FETCH command, the user *should not* use the FOR UPDATE clause in the SQL DECLARE command.

2.2 Repeatable Read with Intent Update

The set of tuples obtained through a cursor, matching the search condition, within a transaction can not be updated by another transaction. Also, this set of tuples can not be read by another transaction unless it is a READ ONLY TRANSACTION and the data has not yet been modified. READ ONLY TRANSACTION is an HP SQL feature under investigation in which updates are not allowed within the transaction. No data can be added to the table that matches the search condition of the cursor through another transaction. Therefore, the same set of tuples are generated if the data is re-read using the same cursor.

```
EXEC SQL BEGIN WORK RR;

EXEC SQL DECLARE c1 CURSOR FOR SELECT * FROM accounts WHERE
                    credit > 1500.00 FOR UPDATE OF credit;

EXEC SQL OPEN c1;

EXEC SQL FETCH c1 INTO :hostvar;

EXEC SQL UPDATE accounts SET credit = :credit*0.1 WHERE CURRENT OF c1;

EXEC SQL CLOSE c1;

EXEC SQL COMMIT WORK;
```

- Index scan: Subexclusive lock on the "accounts" table is acquired by the HP SQL product at open time. When the SQL FETCH command is executed, share locks are acquired for non-leaf index pages, and share subexclusive locks are acquired for leaf index pages and data pages.
- Relation scan: Share subexclusive lock is acquired by the HP SQL product on the "accounts" table at open time. When the SQL FETCH Statement is executed, share subexclusive locks are acquired by the HP SQL product on data pages that match the search condition (credit > 1500.00)
- Concurrency: Only READ ONLY transactions can execute concurrently with the above transaction between the SQL FETCH and SQL UPDATE commands. READ-WRITE transactions can execute concurrently with the above transaction if the READ-WRITE transaction is doing only retrieval on a different subset of tuples than the above transaction and it is using index scan. Once the SQL UPDATE command is executed, the lock on each updated page is promoted from share subexclusive to exclusive. Promotion of a lock from share subexclusive to exclusive does not generate a deadlock.
- Usability: This Isolation Level should be used if the set of tuples to be retrieved will be updated within the transaction. The user should either specify or let it default to the RR option in the SQL BEGIN WORK command and should use the FOR UPDATE clause in the SQL DECLARE command.

2.3 Phantom Read

The set of tuples obtained through a cursor, matching the search condition, within a transaction can not be updated by another transaction; however, it can be read by another transaction. Data can be added to the table that matches the search condition of the cursor through another transaction. *The HP SQL product does not yet support this feature.*

2.4 Phantom Read with Intent Update

The set of tuples obtained through a cursor, matching the search condition, within a transaction can not be updated by another transaction. Also, this set of tuples can not be read by another transaction unless it is a READ ONLY TRANSACTION (updates are not allowed within the transaction). Data can be added to the table that matches the search condition of the cursor through another transaction. *The HP SQL product does not yet support this feature.*

2.5 Cursor Stability

The set of tuples obtained through a cursor, matching the search condition, within a transaction can not be updated by another transaction while the cursor has addressability to the set of tuples; however, it can be read by another READ-WRITE transaction. Data can be added to the table that matches the search condition of the cursor through another transaction.

```
EXEC SQL BEGIN WORK CS;

EXEC SQL SELECT * FROM accounts WHERE
                   credit > 1500.00;

EXEC SQL UPDATE accounts SET credit = credit*1.1
                   WHERE debit = 0;

EXEC SQL COMMIT WORK;
```

In order to explain the type of locks that the HP SQL product acquires on behalf of the SQL SELECT command, we can loosely say that the SQL SELECT command is divided internally into a) open scan, b) several internal FETCH commands, and c) close scan.

- Index scan: Subshare lock on the "accounts" table is acquired by the HP SQL product at open time. When the internal FETCH command is executed, share locks are acquired for the leaf index pages and for data pages containing tuples that match the search condition. No locks are acquired for the non-leaf index pages, and non-exclusive locks are released between internal FETCH commands.

- Relation scan: Subshare lock is acquired by the HP SQL product on the "accounts" table at open time. When the internal FETCH command is executed, share locks are acquired by the HP SQL product for each page that contains tuples that match the search condition. Non-exclusive locks are released between internal FETCH commands.

- Concurrency: READ-WRITE transactions can be executed concurrently with the above transaction. Once the SQL UPDATE command is executed, for each page that is updated, the lock is promoted from share subexclusive to exclusive. Deadlocks can occur in this scheme during the promotion of a lock.

- Usability: This Isolation Level should be used if the set of tuples to be retrieved will *not* be updated within the transaction, and the user will not re-read the data again within the transaction. The user should specify the CS option in the SQL BEGIN WORK command and should *not* use the FOR UPDATE clause in the SQL DECLARE command.

- *Warning:* Since cursor stability holds locks only for the current FETCH command, phantom rows can show up for the above transaction when another transaction inserts or updates tuples containing the credit column that matches the search condition of the above SQL SELECT command. Therefore, if the user tries to re-execute the SQL SELECT command within the same transaction, the user might not get the same set of tuples. More important, a set of tuples previously read may be updated by another transaction and if it is re-read some of the tuples would contain new data. In other words, repeatable reads are not guaranteed by the HP SQL product when the user uses cursor stability. Furthermore, a loss of serializability can occur among transactions since locks are released before the transaction is committed. The execution of a set of transactions is said to be serializable if and only if it produces the same result as some serial execution of those same transactions. To guarantee serializability locks should only be released at the end of the transaction; otherwise, a loss of serializability occurs [2]. For example, a loss of serializability might occurs if transaction T2 is waiting for resource R1 held by transaction T1, and transaction T1 releases resource R1 before T1 commits, then it is possible for transaction T1 to wait for transaction T2 sometime before transaction T1 is committed. The user is also advised not to use the scanned data to update other tables of the data base since serializability among transactions is lost.

2.6 Cursor Stability with Intent Update

The set of tuples obtained through a cursor, matching the search condition, within a transaction can not be updated by another transaction while the cursor has addressability to the set of tuples. This set of tuples on which the cursor has addressability can not be read by another transaction, unless it is a READ ONLY TRANSACTION and the data has not been modified yet. Data can be added to the table that matches the search condition of the cursor.

```
EXEC SQL BEGIN WORK CS;

EXEC SQL DECLARE c1 CURSOR FOR SELECT * FROM accounts WHERE
                        credit > 1500.00 FOR UPDATE OF credit;
```

```
EXEC SQL OPEN c1;

EXEC SQL FETCH c1 INTO :hostvar;

EXEC SQL UPDATE accounts SET credit = :credit*0.1 WHERE CURRENT OF c1;

EXEC SQL CLOSE c1;

EXEC SQL COMMIT WORK;
```

- Index scan: Subexclusive lock on the "accounts" table is acquired by the HP SQL product at open time. When the SQL FETCH command is executed, a share subexclusive lock is acquired by the HP SQL product for the leaf index page, and for data pages that contain a tuple that matches the search condition. No locks are acquired for the non-leaf index pages, and non-exclusive locks are released between two SQL FETCH commands.

- Relation scan: Subexclusive lock is acquired by the HP SQL product on the "accounts" table at open time. When the SQL FETCH Statement is executed, share subexclusive locks are acquired by the HP SQL product for each page that contains tuples that match the search condition. Non-exclusive locks are released between two SQL FETCH commands.

- Concurrency: Only READ ONLY transactions can be executed concurrently with the above transaction between the SQL FETCH and SQL UPDATE commands. READ-WRITE transactions can execute concurrently with the above transaction if they are working in a different set of tuples. Once the SQL UPDATE command is executed, for each page that is updated, the lock is promoted from share subexclusive to exclusive. Since share subexclusive locks are compatible only with intent to share lock, there is not deadlock during the promotion of a lock.

- Usability: This Isolation Level should be used if the set of tuples to be retrieved will be updated within the transaction, and the user will not re-read the data again within the transaction. The user should specify the CS option in the SQL BEGIN WORK command and should use the FOR UPDATE clause in the SQL DECLARE command.

- *Warning:* The cursor stability warning applies to this Isolation Level.

2.7 Read Committed

The set of tuples obtained through a cursor, matching the search condition, within a transaction does not have any locks associated with it, when the data is presented to the user. Locks are acquired during the data retrieval to guarantee that only committed data is read.

```
EXEC SQL BEGIN WORK RC;
```

```
EXEC SQL DECLARE c1 CURSOR FOR SELECT * FROM accounts WHERE
                    credit > 1500.00;

EXEC SQL OPEN c1;

EXEC SQL FETCH c1;

EXEC SQL CLOSE c1;

EXEC SQL COMMIT WORK;
```

- No locks are held when the data is returned to the user regardless of the type of scan. The Read committed option guarantees that the set of tuples is committed and it is not being updated at the instant that HP SQL retrieves the set of tuples. Subshare lock is used on the "accounts" table, and share locks are used on the leaf index pages and data pages; however, all locks are released before the retrieved data is exposed to the user.
- Concurrency: READ-WRITE transactions can be executed concurrently with the above transaction.
- Usability: This Isolation Level should be used if the set of tuples to be retrieved will not be updated within the transaction and the user will not re-read the data again within the transaction. The user should specify the RC option in the SQL BEGIN WORK command and should use the FOR UPDATE clause in the SQL DECLARE command.
- *Warning:* The cursor stability warning applies to this Isolation Level plus data being looked at may be updated. Data should not be updated based on reading with read committed.

2.8 Read Uncommitted

The set of tuples obtained through a cursor, matching the search condition, within a transaction does not have any locks associated with it. No locks are acquired during data retrieval; therefore, the uncommitted data may be read. *The HP SQL produot does not yet support this feature.*

3. HP SQL Isolation Level Implementation

This section describes the implementation of the Isolation Level concept. The major challenge in the implementation of the Isolation Level concept was to answer the following questions:

- Which type of lock should be acquired?
- Which locks should be released before the transaction is committed?
- When the locks should be released?

In order to answer the above questions, the HP SQL Lock Manager was modified by building relationships between the cursor and the lock requests. In other words, the implementation of the Isolation Level concept was only possible with the help of the lock manager, since the control of concurrency among transactions is done by the lock manager.

3.1 Type of lock acquired by the HP SQL product

For each new lock requested, the lock manager allocates a lock control block on behalf of the transaction. The type of lock depends on the Isolation Level that the user has specified in the SQL BEGIN WORK command, and whether or not the user has specified the FOR UPDATE clause in the SQL DECLARE command. Intent to write locks are used for any update command (i.e. SQL DELETE and UPDATE commands). The HP SQL product, depending on the Isolation Level, might release non-exclusive locks before the transaction is committed.

3.2 Which locks to release

In order to know which lock control blocks the Lock Manager has to release, the Lock Manager builds two linked lists to remember what was locked in the previous FETCH command, and what is being locked in the current FETCH command. Both linked lists are chained off the scan control block. A scan control block is created by the HP SQL product whenever a cursor is opened. Therefore, when the Lock Manager is requested to release locks, it gets the locks associated with the previous FETCH command by finding the appropiate linked list pointed by the scan control block.

3.3 When to release locks

A lock request keeps an Isolation Level flag and a counter in each lock control block in order to know when to release a lock. The Isolation Level flag is defined to allow the Lock Manager to release a lock before the transaction is committed. The counter is defined in order to know when to release the lock. If the Isolation Level flag is off, the lock is not released until the transaction is committed. The counter is incremented each time the page is locked on behalf of a cursor (scan control block) and it is decremented whenever the lock manager is requested to release the lock. The lock is released when the counter becomes zero, and the isolation flag is set to true.

4. Benchmark Example

In order to measure the performance gain of an application that uses the concept of Isolation Level, a banking transaction benchmark was designed and implemented. The benchmark simulates a banking system where tellers update bank accounts on-line (done by a foreground process). The generation of reports and bookkeeping work are done in background (done by three background processes).

The benchmark accesses the following tables defined within the data base.

- Accounts Table. It has four columns: account-number, account-name, debit, and credit. A unique index is defined on the account-number column.
- Tellers Table. It has three columns: teller-number, teller-name, and branch-number. A unique index is defined on the teller-number column.
- Branches Table. It has four columns: branch-number, branch-name, debit, and credit. A unique index is defined on the branch-number column.
- History Table. It has seven columns: voucher-number, account-number, teller-number, action, amount, date, and status. A unique index is defined on the voucher-number column and another unique index is defined on date column.
- Voucher Table. It has only one column named voucher-number.

Descriptions of the transactions for the foreground process and background processes are explained in the following paragraphs.

4.1 Transactions for the foreground process

The foreground process executes three types of transactions to simulate a banking system application. The first transaction executes an update of an account in the "accounts" table when money is withdrawn. The second transaction generates a small report on the status of each account. The third transaction increments the voucher number in the voucher table.

4.1.1 First transaction for the foreground process

The client enters the account number, and the teller number and the branch number are entered by the banking system where the client withdraws the money. The execution of the withdrawal transaction involves four steps.

- update to the "debit" field for a given account number in the "accounts" table.

- get a "branch_number" for a given "teller_number" in the "tellers" table.
- update "debit" field for a given "branch_number" in the "branches" table.
- insert a tuple in the "history" table to indicate that an action was done.

4.1.2 Second transaction for the foreground process

The second transaction generates a report for all accounts that are between a range of values provided by the banking system.

4.1.3 Third transaction for the foreground process

The third transaction gets the voucher number from the voucher table and increments it by one (1).

4.2 Transactions for the first background process

The first background process executes two types of transactions. The first transaction uses cursor stability to retrieve data from the "history" table. The second transaction updates the "debit" field on the "accounts" table, and updates the "status" field in the history table.

4.2.1 First transaction for the first background process

The first transaction retrieves tuples from the "history" table which have a value of zero (0) in the "status" field. In this example, a scan on the whole table is done; however, concurrency among transactions is allowed, since cursor stability is used. Cursor stability only holds locks between two SQL FETCH commands, and the BULK SELECT command is broken down internally as a set of calls. Loosely speaking, each call internally executes a FETCH command.

If repeatable read were used instead of cursor stability, serialization among transactions would have happened because a share lock on the "history" table would have prevented writers from accessing the table.

4.2.2 Second transaction for the first background process

The second transaction executes two updates. The first update is done to the "debit" field in the "accounts" table, and the other update is done to the "status" field in the "history" table.

4.3 Transactions for the second background process

The second background process executes two types of transactions. The first transaction uses cursor stability during the retrieval of tuples from the "history" table. The second transaction deletes tuples from the "history" table that match a given voucher number.

4.3.1 First transaction for the second background process

The first transaction scans the whole table to find tuples which have a status value of one (1) in the "history" table. Since cursor stability is used, concurrency among transactions is allowed during the scan of the "history" table. Cursor stability only holds locks between two SQL FETCH commands, and the BULK SELECT command is broken down internally as a set of calls. Loosely speaking, each call internally executes a FETCH command.

If repeatable read were used instead of cursor stability, serialization among transactions would have happened because a share lock on the "history" table would have prevented writers from accessing the table.

4.3.2 Second transaction for the second background process

The second transaction deletes from the "history" table all tuples that match a given voucher number.

4.4 Transactions for the third background process

The third background process executes only one transaction using cursor stability. It produces a report from the "account" table. Cursor stability only holds locks between two SQL FETCH commands, and the BULK SELECT command is broken down internally as a set of calls. Loosely speaking, each call internally executes a FETCH command.

If repeatable read were used instead of cursor stability, serialization among transactions would have happened because a share lock on the "history" table would have prevented writers from accessing the table.

5. Performance Results

Two metrics were used to measure the performance of the HP SQL product. The first one was response time, and the second one was the number of transactions executed per hour.

The following table describes results demostrated by the benchmark when it was run on the on a HP-840 series, with HP-UX release 2.0, 24 Mbytes of memory, and 1 swap space. The benchmark used a 13 secnd delay, and only the third background process was run.

Using only Repeatable Read				
# of users	# of TranX	Ave. Elap(sec)	TranX/Hr	# of Deadlocks
1	50	36199.76	5	0
3	150	35427.99	15	0
5	250	37848.40	24	0

Using Cursor Stability and Repeatable Read with Intent Update				
# of users	# of TranX	Ave. Elap(sec)	TranX/Hr	# of Deadlocks
1	50	1241.38	145	0
7	350	1238.94	1017	0
13	650	1222.57	1914	0
17	850	1282.34	2667	0
23	1150	1280.15	3234	0

Figure 5

The above performance table shows us that the throughput is about 20 times higher when the benchmark uses cursor stability and repeatable read with intent update instead of repeatable read only. Better results have been obtained when the concept of disabling data definition was introduced. Disabling data definition implies that no data definition operations are allowed to execute concurrently with the normal data operations. Performance results using the disabling data definition feature can be found in [1].

6. Conclusions and Future Work

It is clear, from Section 5, that the HP SQL strategy, of allowing the user to specify the isolation level, allows the user to see a tremendous increase in throughput of the system. HP SQL allows the user to control the concurrency of his applications to generate fewer locks and better concurrency, because the user knows the semantics of the data. The Isolation Level concept was implemented to allow the user to control the concurrency of his application programs. The benchmark described in Section 4 was used to show that repeatable read with intent update and cursor stability have provided an increase of throughput 20 times better than the case in which only repeatable read is used.

Online transaction processing applications are characterized by large numbers of users concurrently accessing and updating very large data bases. HP SQL has implemented a flexible synchronization scheme that coordinates the accesses to the data base, and provides a good crash recovery mechanism that preserves the logical and physical consistency of the data base. Section 3 described the modification done to the Lock Manager to achieve a high level of concurrency without perturbing the logical and physical consistency of the data base.

In order to be even more competitive HP SQL is investigating, as a further concurrency enhancement, providing read uncommitted for users who only need physical consistency. Furthermore, the HP SQL product is investigating a feature that allows the user to commit a transaction, thereby releasing both read and write locks, but preserves the scan position for use in the subsequent transaction.

Acknowledgments:

Dora Lee and Edward Cheng were key players in the implementation of cursor stability. Birgit Luebke wrote the benchmark. Birgit Luebke with the help of Edward Cheng were able to collect the performance data. Discussions with Frank Dean and Emmanuel Onuegbe helped me to clarify the concept of Isolation Level. Alex Carlton provided valuable editorial help in the draft version of the paper. The author is also grateful to Sena Palanisami, Scott Walecka, and Sam Prather who supported this implementation.

References

[1] Edward Cheng. Performance in HP SQL. INTEREX, August 1988.

[2] C.J. Date An Introduction to Database Systems Volume II. Addison-Wesley Publishing Company.

[3] J.N. Gray, R.A. Lorie, G.R. Putzolu, I.L. Traiger. Granularity of Locks and Degrees of Consistency in a Shared Data Base. Modeling in Data Base Management Systems. G.M. Nijssen, (ed.) North Holland Publishing Company, 1976 (pages 365-392)

[4] Jim Melton ISO-ANSI (working draft) SQL2. Jim Melton (ed.), April 1988. (pages 23, 24, 190)

[5] Ragaa K. Ishak. Concurrency Control in HP SQL. INTEREX, May 1988.

LINKING DATA PROCESSING AND OFFICE AUTOMATION

Peter O'Neill, Hewlett-Packard GmbH

This paper discusses the problems of being able to effectively-link data processing (DP) installations with computing facilities in an office environment (OA). It is, of course, not a new scenario, as can be illustrated by the following story.

In 1976, an accountant at an automobile manufacturer in Great Britain moved into a job where he had to report the sales market share, on a DAILY basis, to upper management. The procedure already set up was that an external system sent data to the company system, which then output a daily report. This daily report was a little cryptic and needed some interpretation. The raw data was then combined with analysis and typed up (cut and pasted) by the secretary. Then the report would be hand-delivered to each manager (well, at least into the "IN" basket!).

A pretty basic informational task you'd think: the provision of data to management on a regular basis in a readable form. In time, improvements were made to the process by using a report writer to set up a better report. And an Apple IIe (remember them?) was used to maintain a spreadsheet from which the final daily report could be generated automatically. It was still delivered by hand to each office though.

No doubt, you have similar stories to tell about information transfer in your organizations. And, no doubt, things have improved over the last 12 years. Or have they? What does Data Processing and Office Automation mean in most companies today?

DP: THE IMPLEMENTATION OF BUSINESS TRANSACTIONS

Data processing can be defined as the implementation and maintenance of an application, either packaged or bespoke. These applications work at a transactional level, produce regular production reports and deal with decisions such as "IF credit-limit > ord-total THEN accept ELSE reject".

Database(s) used in the applications are designed with these application decisions in mind. Usually, some reporting considerations are also reflected in the database design. However, that is restricted to the production reports and the sort of reporting and data presentation that business people require is not a priority. In fact, at the time of application design, these needs may not even be known. A product manager at HP, for example, uses statistics from databases that were designed many years ago and are, for the type of analysis now required, almost inaccessible.

In summary, the data being maintained and the decisions being made in the DP environment are purely OPERATIONAL. And the operational data is mostly limited to single applications. But there are other information needs within a company. These can be described as TACTICAL and STRATEGIC. These three types of data are described in Figure 1.

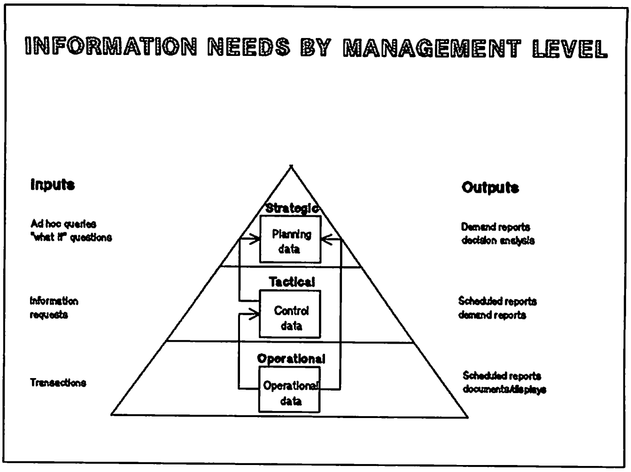


Figure 1.

An example of tactical data, using the credit limit example from above, is a report of how much of a credit limit is being utilized by a dealer.

A related strategic decision would be the evaluation of possible new credit limits to reflect a pending new product which will cost 30% less than the current one.

INFORMATION IS A RESOURCE!

The competitiveness of a company is directly related to its ability to make tactical and even strategic decisions. For example, a company that depends heavily on the market price for a certain commodity must be able to react very quickly if external factors, such as politics, affect the price. Currently, it probably requires several months to collect enough data to support a strategic decision -- expect this time to be reduced to a few days by the 1990s (which doesn't mean, however, that a company will make 2 or 3 strategic decisions per week!).

The battle to reduce the time taken for operational decisions can be considered as being over, at least on minicomputers. It occurs when companies move from batch applications to online. And at the mainframe level, there is now a strong trend to implementing On-Line Transaction Processing (OLTP) for applications that were previously only available in batch.

Tactical and strategic decisions require analysis based on operational data PLUS possibly data from other applications, or even from external sources. Linking DP and OA should address THIS need.

When examining the types of tools available today for DP professionals, you will find that they concentrate mostly on the presention of operational data. These are mainly report-writer tools, although some of the tools also have limited graphics capabilities.

However, with few exceptions, these report-writer tools are really batch list-writer tools; they process operational data into a printed listing which is then used a source for manual analysis. Some companies callthese listings "tabs", they are printed on computer paper and can be hundred of pages long. You can even buy special binders to keep these tabs tidy.

Expense reporting and controlling still means, in most companies:

1) Printing a report
2) Splitting the "tab" into the separate cost center details (using a 12-inch rule perhaps)
3) Putting the individual reports into envelopes
4) Mailing the reports to the managers via an in-house postal service.

Sometimes, separate reports are printed within one job to save the manual splitting task.

Usually , the listings report just one application at a time -- the tools are usually purchased as part a toolset for the application package. And another package, written in another toolset requires separate reports. Only a few of the tools can combine data from several applications on one report.

OA: THE PRESENTATION OF DATA FOR TACTICAL AND STRATEGIC DECISIONS.

Office Automation, in the context of this paper, is now quite easy to define. It is the presentation of data for strategic and tactical decisions. Important in this respect is the following:

* The ready access to operational data,
* The ability to combine other data with this operational data
* The capability to present the consolidated data.

Hewlett-Packard's view is that the integration and presentation of such data is most effective when accomplished on the personal computer. Many companies have set up "information centers" using PCs and PC software to produce spreadsheets and graphics. And HP is undoubtedly one of the leading vendors in linking PC users to databases through the HP INFORMATION ACCESS product.

However, most companies' operational databases are proving too complicated for end users to be able to get the information they require. One hears comments like:

* "The data is too dirty for end users to use"
* "End users are hogging valuable resources"

from MIS staff involved in such projects.

By "dirty data" they mean that data is sometimes stored in a form which, although perfectly understandable to a programmer, does not make much sense to end users reporting from it.

One example is a user who, when analysing product sales statistics from a file containing delivery data, obtained sales results far larger than expected. It took much research to find out that this was because deliveries of promotional items, such as pens sold at a large discount with every third unit sold, were included in the sales statistics.

Another example is a user creating a report which then printed various date items in "days since 1990"!

And of course, a common critisism of end user access to operational databases is that the user sometimes sets up queries that then take up far too many computer resources.

LINKING DP AND OA

So, some companies are now coming to the following conclusions about implementing a better link between DP and OA:

1) CHANGE THE USE OF REPORT WRITERS
Report writers could be used not only for production reporting but to reduce operational data to a form that can be of direct use in an office.

   It is for this reason that some report writers can now output their data directly in spreadsheet formats. As well as generating listings of data, report writers should be able to FILTER operational data for office use. Not only filter but also TRANSFORM the data -- the report writer should be able to process "dirty data" and provide several file formats.

2) USE ELECTRONIC MAIL SERVICES
Electronic mail (E-MAIL) should be used to transmit data as well messages. It is an obvious extension of such a system with two advantages: once an E-Mail service is established, then so is the necessary networking; and users have a ready-built interface to be able to handle deal with the data.

   A popular enhancement request for HPDESK, for example, has been programmatic access to the product. One of the reasons that many customers request this is to be able to link their own programs to the product and send data around the network.

3) LINK DP AND OA TOOLS
The DP and OA tools should be able to talk to each other.

There are report writers for programmers and applications developers. And there are report writers for end users. The two product types are similar but they do have distinct differences in style and capabilities. But, ideally, they should be able to communicate data to each other. The data output by production report writers should be accessible from the preferred office tools.

Hewlett-Packard has now addressed these concerns and implemented the following product enhancements:

* HP BUSINESS REPORT WRITER (HP BRW)
  Under the project name, "HP BRW -- OFFICE", HP BRW has been enhanced to link it closer to both E-MAIL services on the HP3000 and to HP's PC report writer. These enhancements are described in more detail on the following pages.

* HP DESK
  HP DESK intrinsics are now available to enable you to better integrate your applications with the services. These intrinsics are described in a separate paper being presented at this conference and they have been implemented in HP BRW as part of the HP BRW -- Office project.

  In addition, HP DESK has been enhanced to allow the browsing of BRW reports within the IN TRAY, with all the scrolling features available today on HP BRW's online review screen.

* HP INFORMATION ACCESS
  HP INFORMATION ACCESS has been enhanced to be able to read files which were output by HP BRW.

INTRODUCING " HP BRW - OFFICE "

HP BRW is HP's report writer for DP staff and has been installed at over 1,500 sites since its release over 2 years ago.

HP BRW can be used to produce reports across many applications and its data access, processing and formatting capabilities have ensured that it is used for tasks that were only possible, till now, with a programming language like COBOL. It also defines high-performance reports because of its report tuning features.

Now functionality has been added to HP BRW in the following areas:

1) BRW-DESK
   Through this add-on product, HP BRW can be instructed to deliver a report ELECTRONICALLY, using HPDESK to one or more IN TRAYs. The HP DESK user(s) can read the report directly from the IN TRAY as well as print it out like any other message.

   The HP DESK addresses are defined as part of the BRW report via a new function in HP BRW. A single report can also be split electronically, at a certain sort level, and distributed to a list of HPDESK users.

2) New Output File Types
   HP BRW can output files in a list of possible formats including those of PC spreadsheet programs or graphics packages for direct use in these OA tools.

3) Intermediate Report File (IRF)
   Data can now be passed between HP BRW, the DP report writer, and HP INFORMATION ACCESS, the end-user report writer on the PC.

   This is because HP BRW's Intermediate Report Format (IRF) can now be read by HP INFORMATION ACCESS. The converse is also true: files can be output from HP INFORMATION ACCESS in IRF formet which can be read by HP BRW. This means that HP BRW reports can include data from foreign database systems via HP INFORMATION ACCESS CULLINET LINK.

4) Complete Reporting System
   HP BRW and HP INFORMATION ACCESS now share the same dictionary. This fact, along with the possibility to move data between the products, means that the purchase of HP BRW and HP INFORMATION ACCESS serves as a complete reporting solution:

   * HP BRW for the MIS area
   * HP INFORMATION ACCESS for the office environment.

The following pages list just some of the implementations you could now consider with the addition of these products.

LINKING DP AND OA -- 1 (see Figure 2)

An expense report is specified, with HP BRW, within the regular month-end routines. This report is sorted by cost center and shows the expenses versus budget. Automatically, each individual cost center listing is separated out and sent to the reponsible manager, with a carbon copy to the accounting manager. Via HP DESK, the CC manager can review the expenses report online and/or print it out as with any HP DESK message.

An exception list of all cost centers with greater than 20% variance is also collected and sent to the accounting manager, from the same report.

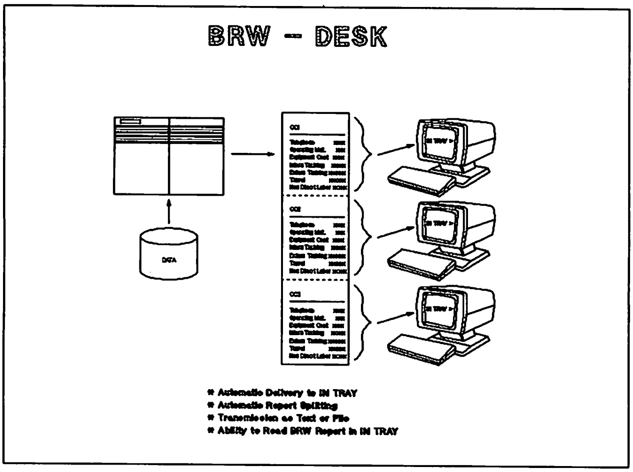


Figure 2.

LINKING DP AND OA -- 2

An orders report is run, with HP BRW, at the end of each month, listing the units and revenues sold for various products. The report is split electronically, by HP BRW, into individual product reports and each component is sent, via HP DESK, to the responsible product manager. The data is sent in DIF format which means it can be imported directly into spreadsheet and graphic packages on the product manager's PC.

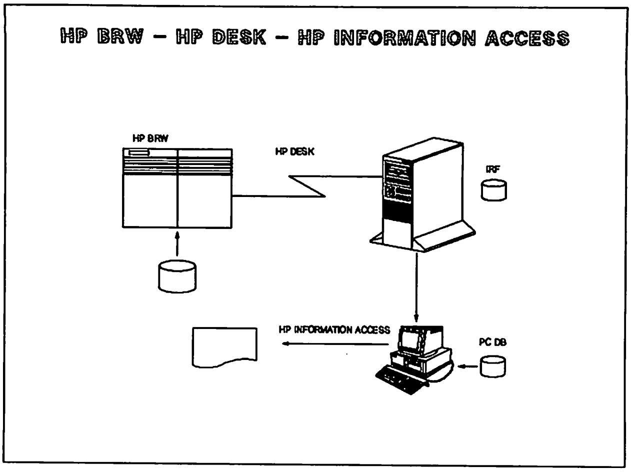


Figure 3.

LINKING DP AND OA -- 3 (see Figure 3)

A business analyst needs to produce a report combining the latest order statistics with a some market analysis data. Order statistics are received on a regular basis, in Intermediate Report Format (IRF) via HP BRW and HPDESK. The market data is held on a PC database. Using HP INFORMATION ACCESS, the business analyst produces the required report in a matter of moments.

LINKING DP AND OA -- 4

In the US, HP produces order status reports, via HP BRW, on a monthly basis which are sent to a network of its major customers via HP DESK.

LINKING DP AND OA -- 5

The accounts payable department prints payment instructions for the company bank on a weekly basis. These payment instructions are output via a HP BRW report and sent electronically to the bank using HP DESK and HP TELEX.

LINKING DP AND OA -- 6

The financial budgets for a new year are due. The Controlling department run an HP BRW report which lists the previous years data plus a recommended multiple for the new year. The report is output directly in a spreadsheet format and sent to all managers via HP DESK plus HP ADVANCE MAIL. The file is now imported into a spreadsheet program and the new budgets are created online.

CONCLUSION

These are some of the possibilities with the "HP BRW -- OFFICE" features. We are confident that all of these features mean that a combination of HP BRW, HP DESK and HP INFORMATION ACCESS is now the ideal toolset to move data between production applications and an office environment.

The above functionality is available in Fall, 1988 and will definitely help you to forge better links between data processing and office automation.

Concurrency Control In HP SQL

Ragaa K. Ishak

Distributed Data Management Lab
Hewlett-Packard
19447 Pruneridge Avenue
Cupertino, CA 95014

ABSTRACT

Hewlett-Packard's proprietary relational database management product HP SQL provides data integrity, and high concurrency between users. In any system that allows a high number of concurrent users, there is always the possibility of two or more users trying to update the same data simultaneously. HP SQL deals with the issue of concurrency and data integrity by using locks to restrict a user's access to data. The use of locks, which is handled by a lock manager, is fundamental to HP SQL's ability to ensure data integrity.

This paper describes the functionality and internal design of the locking mechanism in HP SQL. This paper is aimed at readers who want to write better applications using the full strength of HP SQL and readers who want to write diagnostic tools. This knowledge is crucial for designing high throughput applications and useful tools.

This paper assumes that the reader is familiar with HP SQL, however starters may still find it informative. The reader is assumed to be familiar with HP SQL concepts and specifically with the HP SQL structures such as tables, rows, indexes, DBEfiles, and DBEfilesets.

1. Introduction

HP SQL can support up to 240 concurrent transactions. Concurrency in HP SQL is controlled by using locks. Implicit as well as explicit locks are supported by HP SQL. Different types of locks are used depending on the type of table created, lock commands, and options specified when a transaction begins. Locks are set on user tables, as well as internal tables. The locks involved in SQL operations are described in section 3. Compatibility rules and conversion rules tell us what operations can be concurrent. The compatibility rules for locks are discussed in section 4. The conversion rules for locks are discussed in section A look is taken behind the scenes at how concurrency is managed. The internal design of locking inside HP SQL is described in section 6. The different locks set during the different SQL commands are overviewed in section 7. Finally, the presented information will be used in drawing some conclusions pertaining to performance in section 8.

2. Overview of HP SQL Structure, Role of DBCORE

HP SQL consists of two major subsystems, SQLCORE and DBCORE. Discussion of other components of HP SQL is outside the scope of this paper. HP SQL uses a lock manager to synchronize access to data and manage locks. The lock manager in HP SQL is part of DBCORE.

DBCORE performs the basic DBMS functions of data definitions, data access, transaction management, concurrency control, logging and recovery, and accounting. In this paper, we are concerned with the concurrency control services that DBCORE provides with HP SQL.

SQLCORE translates SQL commands into an access plan. The access plan contains a sequence of DBCORE calls to define, store, and retrieve data. SQLCORE uses an optimizer to generate the most effective plan to retrieve data. The access plan consists of one or more DBCORE calls.

3. Lock Types

In this section, the lock types supported by HP SQL are described. Lock granularity, and intention locks are introduced.

a. Lock Granularity

To synchronize access to data, READ locks and WRITE locks are set. The type of table created determines the locking unit during read and write operations. For example a table of type PRIVATE is locked exclusively at the table level whenever accessed for read or write. A table of type PUBLICREAD is locked at the table level when doing a write operation and also locked at the table level when doing a read operation. A table of type PUBLIC is usually locked at the page level when doing read or write operations. READ locks can be shared but WRITE locks are exclusive. In summary, the locking unit on a user table is a page or the table itself.

Throughout this paper a lockable unit will also be referred to as an *object*. An object, a locking unit, has a *granularity* level. A table in HP SQL consists of one or several pages. The granularity of a page, then, is finer than the granularity of a table. In the same way, the granularity of a row is finer than the granularity of a page.

b. Intention Locks

When an object needs to be locked, several locks are set. When a lock is set on a relatively finer granularity object say a page, then an intention lock is set on the coarser granularity objects containing this object such as the table containing this page.

To illustrate the need for intention locks, consider the following scenario. Suppose that a given transaction T wishes to process some large table A, and also T requires A to be stable, i.e. it cannot tolerate changes in A by a concurrent transaction. Suppose also that T itself does not wish to make changes in A either. T can achieve the desired stability in A by obtaining a share lock on A. On receipt of T's Share lock request, HP SQL must be able to tell whether any other transaction is making changes to A, i.e., whether any other transaction already has an eXclusive lock on any page in A. If so, then T's request cannot be granted at this time. How can HP SQL detect this conflict ?

It is undesirable to have to examine every page in the large table A to see whether any one of them is X locked by any transaction, or to have to examine all existing eXclusive locks to see whether any one of them is for a page in A. Instead, intent locking is used. The need for intent locking is even more obvious when locks are set at the row level. *Intention* locks are used to speed up compatibility checking and to reduce deadlocks.

1) To lock a page in table T for read (page P)
- lock table T with mode (IS)
- lock page P with mode (S)

2) To lock a page in table T for write (page P)
- lock table T with mode (IX)
- lock page P with mode (X)

3) To lock a page in table T for read/potential write (page P)
- lock table T with mode (IX)
- lock page P with mode (SIX)

4) To lock row for read (row T)
- lock table containing T with mode (IS)
- lock page containing T with mode (IS)
- lock row T with mode (S)

5) To lock row for write (row T)
- lock table containing T with mode (IX)
- lock page containing T with mode (IX)
- lock row T with mode (X)

Intention Locks Generated

Figure 1

Figure 1 lists all the locks that are generated on an object during different types of operations. Note that row level locks are not currently supported on user tables.

c. Lock Summary

IS – Share lock at a finer granularity level

IX – EXclusive lock at a finer granularity level

S – Share lock at this level

SIX – Share lock at this level with intention of updating

X – EXclusive lock at this level

HP SQL Locks

Figure 2

Figure 2 lists all the locks supported by HP SQL. In summary, HP SQL uses five types of locks: Share (S), eXclusive (X), Intent Share (IS), Intent eXclusive (IX), and Share Intent eXclusive (SIX). An SIX, set by a transaction on an object allows multiple readers at finer granularities of this object, and disallows writers except the first transaction. More on the usage of SIX locks can be found in the section "compatibility between different locks". Note that the locks IS and IX cannot be set by the user. IS and IX are set internally by HP SQL to facilitate lock management. Row level locks are set on internal tables as described in a later section.

4. HP SQL Locking Modes

The user can control the concurrency level in HP SQL in three ways. First, when the table is created, a table type can be specified. Second, when a transaction is started, an isolation level can be specified. Third, in the course of a transaction, an explicit lock command can be specified. Implicit locks take effect when no explicit lock commands are specified.

a. Table Types

One of three following options can be specified when a table is created:

I. CREATE TABLE PRIVATE which implies one user at a time.

II. CREATE TABLE PUBLICREAD which implies one writer or multiple readers allowed to access the table concurrently.

III. CREATE TABLE PUBLIC which implies multiple writers and multiple readers allowed to access the table concurrently.

Table Type	Read Locks	Write Locks
Private	Exclusive Table (X)	Exclusive Table (X)
Publicread	Share Table (S)	Exclusive Table (X)
Public	Share Table (IS) Share page (S)	Intent Exclusive Table (IX) Exclusive page (X)

Implicit Lock Summary Table

Figure 3

Figure 3 shows the implicit locks that are generated for the different types of user tables. The locks shown in the table are the locks generated on the user table itself. The locks shown are for read and write operations. Note that for a PUBLICREAD table, a Share lock is set on the table for read operations. Since a write lock on such a table is eXclusive, then individual page Share locks are not necessary. The combination of Share locks and eXclusive locks at the page level is not allowed.

In the case of PUBLIC tables, the locks shown during read operations assume an index scan. During an index scan, the address of the requested data is obtained from the index. The pages on which the data reside are accessed directly and Share locks need to be set only on these pages. If the index is not used, then the whole table is scanned to find the data. As a result, a share lock is set at the table level.

In addition to the locks mentioned in the table for read and write, SIX (Share with Intent eXclusive) locks can be placed at the page or table level. When read with intent to update operations are requested, SIX locks are set in HP SQL. For example the SQL command pair "DECLARE CURSOR ..FOR UPDATE, OPEN CURSOR.." is an intent to update operation. The SIX lock is also set at the table level with the command LOCK TABLE ..IN SHARE UPDATE MODE.

b. Explicit Locks

Explicit lock commands can be specified in HP SQL. The advantage of specifying a lock command on a table is to reduce the number of locks generated, and therefore reduce the overhead of managing these locks. An explicit lock command such as LOCK TABLE ..IN EXCLUSIVE MODE will place one lock on the table, thus eliminating multiple locks on several pages and eliminating intention locks. The LOCK TABLE ..IN EXCLUSIVE MODE reduces the concurrency on the table to one transaction but gives this transaction the exclusive access to the table, therefore inproving the performance of this transaction. An explicit lock on a table can also be used to prevent deadlocks. A table can be locked explicitly as EXCLUSIVE, SHARE, or SHARE UPDATE.

c. Isolation Levels

The user can also control concurrency by specifying an *isolation level* when issuing the BEGIN WORK command. An isolation level is a consistency level for the data seen by a transaction. An isolation level is the degree of isolation a particular database access has from other attempts to access the database. The isolation levels currently supported by HP SQL are *repeatable read* and *cursor stability*. With the repeatable read level of consistency, also the default, HP SQL ensures that if data are repeatedly read in a transaction, the same data will be seen. To achieve this consistency level, READ locks, as well as WRITE locks of course, are kept until the end of the transaction. Data seen by transaction T1 cannot be updated by transaction T2 until T1 is finished.

With the cursor stability level of consistency, data seen is guaranteed to stay the same only as long the data is being addressed. If data is read again in the same transaction, seeing the same data is not guaranteed. With cursor stability, more concurrency can be achieved, since transactions keep their READ locks for shorter periods of time. Note that, the cursor stability consistency level is also achieved with the repeatable read consistency level, but the repeatable read consistency level is not achieved with cursor stability.

A higher performance is always very desirable in a DBMS. One of the major factors contributing to performance is the concurrency level. A higher concurrency level can be achieved with a lower consistency level, i.e., isolation level. As a result, there is a great potential of a performance increase, when cursor stability is used over repeatable read, in a multi user environment. In a single user system, cursor stability reduces performance.

Isolation levels, lower than repeatable read and cursor stability can be used to increase concurrency. *Read committed* and *read uncommitted* are lower isolation levels, currently under investigations for future releases of HP SQL. With the read committed isolation level, data seen is guaranteed to be committed, but not the same while it is being addressed. READ locks are kept, only while the data is being fetched from the database. With the read committed isolation level, more concurrency can be achieved than in cursor stability, since transactions keep their READ locks for shorter periods of time.

With the read uncommitted isolation level, the DBMS ensures that no updates are lost, i.e., no one transaction can write over uncommitted changes of another transaction. However, one transaction can read uncommitted changes by another transaction. At this level, reading committed data, cursor stability, and consequently repeatable reads are not guaranteed in a transaction. On the other hand, a read only operation should minimally wait for a lock and minimally effect waiters.

d. Commit/Keep Cursor

Commit/Keep Cursor is an enhancement under investigation for future releases of HP SQL. Currently all locks, READ and WRITE, are released, and the cursors set to null when a transaction ends, i.e., COMMIT WORK or ROLLBACK WORK are issued. With the Commit/Keep cursor option, implicit locks set to guarantee the consistency levels repeatable read and cursor stability may be retained (Kept) across COMMIT WORK commands. A cursor position is kept across COMMIT WORK commands.

5. Compatibility Between Different Locks

The compatibility between the locks tells us what operations on common data can be concurrent. Figure 4 shows the compatibility matrix between the five different locks that can be placed on objects in the database. As the table shows, some locks are compatible, and some are not. For example, two Intent eXclusive(IX) locks are compatible. An Intent eXclusive (IX) lock is placed on a user table of type public when the user is updating a page with an eXclusive lock. Another user may update another page with an eXclusive lock in the table and share the Intent eXclusive (IX) lock on the table. A Share (S) lock compatible with a (S) lock means that two user transactions can read the same pages in a table at the same time, and so on...

	IS	IX	S	SIX	X
IS	Y	Y	Y	Y	N
IX	Y	Y	N	N	N
S	Y	N	Y	N	N
SIX	Y	N	N	N	N
X	N	N	N	N	N

Lock Compatibility Matrix

Figure 4

a. SIX Lock Implications

Figure 4 shows that an SIX lock is compatible only with an IS lock. This compatibility implies that if an SIX lock is set on a user table, then only readers, with Share locks, are allowed at the page level. Remember, that a Share lock at the page level generates an IS lock at the table level. No writers are allowed to this table at any level. So, in effect, an SIX lock is weaker than an eXclusive lock at a certain level, because it allows readers at finer granularities. An X lock is, of course, exclusive. On the other hand, an SIX lock is stronger than an IX lock because an IX lock set at a certain level, allows writers in addition to readers, at finer granularities. If an SIX lock is set by a transaction, on a page in a user table, then it is equivalent to an eXclusive lock on the page. Remember, that currently, row level locks are not allowed on user tables. If row level locks were allowed on user tables, readers will be allowed at the row level when an SIX is set at the page level. To summarize the effects of an SIX lock, we can assert that it is stronger than an IX lock and weaker than an eXclusive lock at a certain level. An SIX lock at the finest granularity, such as row, is meaningless.

When an SIX lock is set at the page level, an IX lock is set at the table level. If an SIX lock is set, then when the transaction updates, the SIX lock is upgraded to eXclusive.

b. Wait For Lock Protocol

If a lock request is not compatible with an existing lock on the same object then the user transaction will wait until the lock can be granted.

One important fact to be aware of, is that even though the compatibility rules allow two concurrent share locks on the same object, a second share lock might not be instantly granted for another transaction. If a Share lock is granted, an eXclusive lock request is waiting first in line, a second Share lock request will wait second in line.

The other fact is that, if a first transaction T1 is granted a Share lock, a second transaction T2 is waiting for an eXclusive lock, and the first transaction decides to convert from Share to eXclusive, then the converter is given priority for the eXclusive lock. If a Share lock is already granted, then the converter will have to wait. A reader who intends to update may request an SIX lock instead of Share to disallow granting Share locks on the same object, and eliminate the wait for lock conversion.

The application developer should keep the above protocols in mind while designing an application.

5. Conversion Between Different Locks

When changing the type of operation on an object such as from read to write, a lock might need to be converted. There is also the case where an explicit lock needs to be specified to overrule the implicit locks.

	New Mode				
Old	IS	IX	S	SIX	X
IS	IS	IX	S	SIX	X
IX	IX	IX	SIX	SIX	X
S	S	IX	S	SIX	X
SIX	SIX	SIX	SIX	SIX	X
X	X	X	X	X	X

Lock Conversion Matrix

Figure 5

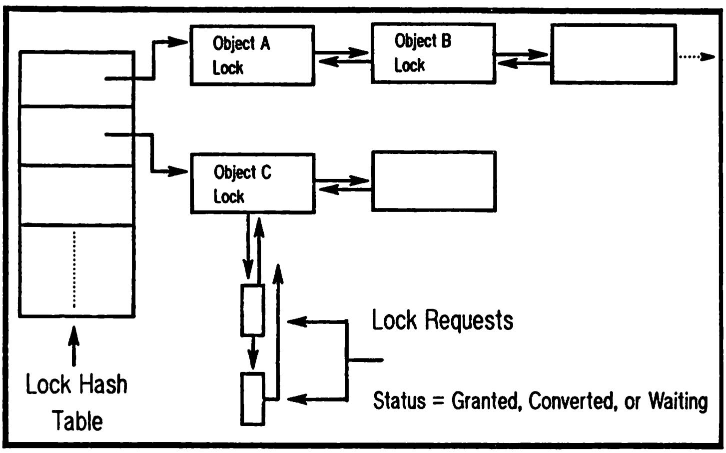
If a lock is already granted, then it can only be upgraded to a stronger lock. Figure 5 shows the lock conversion matrix. Note for example that an IS lock can be converted to an eXclusive lock, but not vice versa. As a result of the conversion rules, an explicit lock command can overrule the locks set on the table only if it is stronger. For example if a table is created as private and the command LOCK TABLE SHARE is issued, a read operation still effects an eXclusive lock on the table.

6. Concurrency Management

In this section, the design of the lock manager in HP SQL is presented.

a. Objects, Locks, And Lock Requests

Figure 6 shows the relationships between objects, locks, and lock requests. When the database is first started with the SQL commands START DBE NEW, START DBE, or CONNECT with autostart option on, a lock hash table is created. The size of the hash table is dependent on the number of transactions specified when the database is started. A lock is implemented as a control block. The hash table keeps lists of locks. Each object, i.e., a table, a page, a row, has a unique identification number in the database. This unique identification number is used to hash the object to be locked, and assign a lock to it. Each request to lock an object is a lock request. A lock request is also implemented as a control block. A lock request is associated with one and only one lock. However, there can be multiple requests to lock the same object.

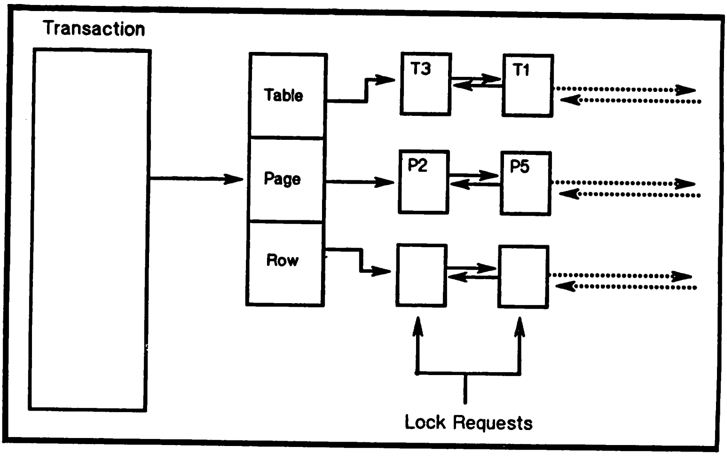


Lock Management

Figure 6

b. Transaction And Lock Requests

Figure 7 shows the relationship of a transaction to its lock requests. For each transaction, three lists of lock requests are kept. The first list is a list of lock requests for locks on tables. The second is a list of lock requests for locks on pages. The third is a list of lock requests for locks on rows. The status of a lock request can be either granted, waiting, or converting. For each scan, i.e., "FETCH" command, a list of READ lock requests granted is kept. In the case where cursor stability option is specified, this list is used to release the locks acquired during a previous "FETCH".



Transaction Lock Requests

Figure 7

c. Processing Of Lock Requests

I. When a lock is requested on an object, the lock requests issued by this transaction are checked first in the sequence table, page, and row.

II. If some lock request by this transaction already exists for the object, then the old lock request is converted according to conversion rules.

III. If the lock request for this transaction is not found, then compute a hash location for the object to be locked and the hash table is checked for it.

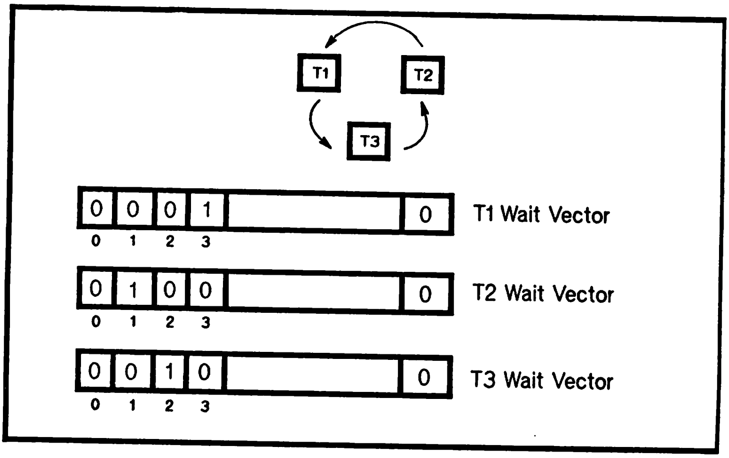
IV. If a lock for this object is not found in the hash table, a new lock is allocated and a lock is requested on the object.

V. If a lock is found, then it is determined if the lock requested is compatible with the existing granted lock. If compatible then it is granted, else place the new lock request in wait status.

VI. If a lock needs to be converted to a stronger lock, it is added to the wait list for this lock. Requests for read and write are given the same priority in the wait list. The lock wait vectors of waiting transactions are updated.

d. Deadlock Detection

For each transaction, a wait vector is assigned. A bit in the vector is assigned for every transaction in the database. If a transaction waits for a lock held by another transaction then the corresponding bit of the other transaction is set in the first's wait vector. HP SQL checks for deadlocks whenever a lock request causes a wait. Figure 8 shows an example of wait vectors for three transactions in the database. T2 is waiting for T1. T1 is waiting for T3. T3 is waiting for T2.



Transaction Wait Vectors

Figure 8

The wait vector is updated for all waiting transactions each time the status of locks changes, e.g. a lock is released. A deadlock with part of the cycle outside the HP SQL system cannot be detected. This might be an obvious assertion, still, application complexity increases the potential of such scenarios.

There are various deadlock avoidence protocols that can be used. For example, a scheme is to request all locks in advance. Using this scheme is not possible where a high level of concurrency is required and one of the following conditions exists:

I. A transaction T may be unable to identify row R1, and hence unable to lock it, until it has examined R2.

II. The set of lockable objects is very large, consisting possibly of thousands of pages, but it also changes dynamically.

III. The precise number of lockable objects is not known in advance.

IV. Records are addressed not by name, but by content, so that it cannot be determined until execution time whether or not two distinct requests are for the same object.

7. Locking Activities During SQL commands:

In previous sections we learned about lock types, locks set during different operations, locks set on user tables, concurrent locks, and lock management internals. In this section, a closer look is taken at all the locks that are generated during different operations. To complete the picture, all the tables involved in the DBE are overviewed.

a. Tables In The DBE

There are three types of tables involved in a Data Base Environment (DBE):

I. The tables that the user creates and uses to store data. These tables are referred to as the user tables. User tables are locked at the page level or the table level.

II. The SQL catalog tables that contains information about the user's DBE, such as names of tables, column names, column types, authorization groups, etc.. The SQL catalog tables are locked at the page level.

III. The DBCORE internal tables contain low level basic information about the data.

When a Data Base Environment (DBE) is first created four DBCORE tables are also created. They contain information about tables, DBEfiles, DBEfilesets, and indexes in the DBE respectively. The information contained in these tables is the minimum information needed to store and retrieve data. For example, these tables do not contain names for tables or columns, but instead they contain identification numbers. SQLCORE translates table names into identification numbers and column names into positions within a row. The SQL catalog tables keep all the information necessary to provide the user with a high level interface to data. The DBCORE tables are locked at the row level. READ or WRITE locks are set on these tables at the row level during the various operations involving data access.

b. START DBE Command

When the START DBE NEW command is issued, the DBCORE tables are first created, then the SQL catalog tables are created. Each SQL catalog table is created as a separate transaction, surrounded by BEGIN WORK and COMMIT WORK commands internally. No locks are held after a START DBE command is issued.

With any START DBE command, the lock control blocks are allocated, formatted, and the lock hash table is initialized.

c. Data Definition Language Commands

A Data Definition Language (DDL) commands causes a change in the structure of data. With a change in the structure of the data, the DBCORE tables and the SQL catalog tables need to be updated. Exclusive WRITE locks need to be set on one or more internal tables. Since READ locks can be shared but WRITE locks are not, DDL commands tend to reduce the concurrency in the DBE considerably. To maximize the concurrency in a production environment, DDL commands need to be scheduled.

To improve performance, HP SQL supports an option where DDL commands can be disabled. When the user chooses the option of disabling DDL, HP SQL can keep executable sections in memory, eliminating the time spent in repeatedly bringing sections from the SQL catalog. When DDL commands are enabled, sections can be invalidated. A section to be executed needs to be brought from the catalog each time so that it can be revalidated if neccessary.

d. Data Manipulation Language Commands

Data Manipulation Language commands (DML) change the data itself but not the structure of the data. The DBCORE and SQL tables are locked in Share mode to prevent concurrent DDL, i.e. concurrent data restructuring. Share page locks are set on the SQL catalog tables and Share row level locks are set on the DBCORE tables.

e. UPDATE STATISTICS Command

An UPDATE STATISTICS command on a table causes a scan of the DBEfileset containing the table to find the pages belonging to it. The number of pages are counted. Each page belonging to the table is then scanned and the size of each row is checked. The average size of a row in the table is then computed. Similar index information is computed for each index on the table. To guarantee the consistency of data during this operation, exclusive locks are set on the DBCORE tables for table and index information. Exclusive locks are set on the SQL catalog tables because they are updated. No other concurrent operation on the table is allowed during an UPDATE STATISTICS command. To allow other transactions to access the table, a COMMIT WORK needs to be issued. SQLCORE uses the statistics on a table to generate optimized access plans to retrieve data.

f. CHECKPOINT Command.

A CHECKPOINT command does not acquire any locks. All updated data pages are written to disc so access to the data buffer pool is prevented. Concurrency is reduced considerably during this operation, while buffers are written to disc and log file space is reclaimed, but no locks are held.

A complete list of the type of locks set on the SQL tables is in the DataBase Administration manual.

9. Conclusions Pertaining To Performance.

Different applications have different requirements. As a result different techniques may be used. Following are some guidelines for performance improvement. If no specific application need is mentioned then the guideline applies for most applications.

I. Keep transactions short, especially if they use DDL commands. As mentioned in the previous section, DDL commands effect exclusive page locks on the SQL tables. Concurrency is then reduced considerably. If DDL commands can be scheduled, then use the DDL DISABLED option.

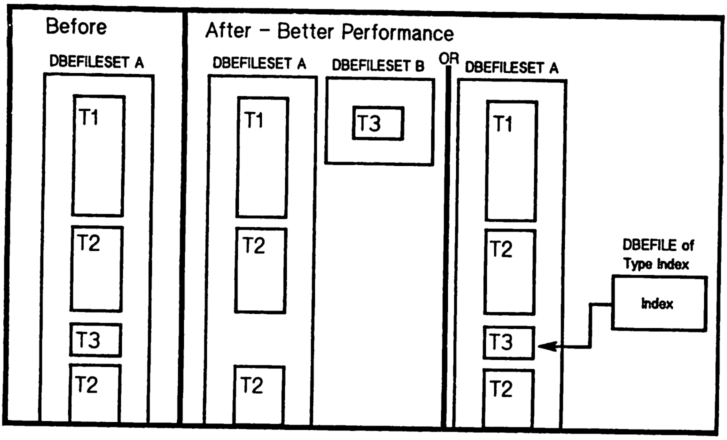
II. Specify SHARE UPDATE locks. This kind of lock increases concurrency on a table when a transaction needs to update many pages in the table and still allow other transactions to read.

III. Use the Cursor Stability option. Your READ locks are released as you read other pages so concurrency is increased. Your read performance decreases slightly to release no longer needed READ locks.

IV. The following guideline was learned from witnessing a benchmark application. The original performance was unsatisfactory. It turned out that the application was accessing a small table very often. Almost all transactions accessed this small table. The small table was stored in a DBEfileset with many other large tables. The left side of Figure 9 shows the "Before" placement of the small table. To access the table, a DBEfileset scan is performed to locate the page(s) owned by the small table. This is a slow process considering that there are alternatives to increase the concurrency in such a system. The right side in Figure 9 shows the "After" alternatives. The small table can be placed in a separate DBEfileset to speed up the process of finding it. The second alternative is to create an index for this table. The index contains the exact location (pages) owned by this table. For faster access, the index can be placed in a separate index type DBEfile.

V. Reduce I/O while holding locks. Terminal and disc I/O may consume unexpected amount of time.

VI. To increase the performance of an individual transaction, use locks at the table level. A lock at the table level reduces the overhead of intention locks, locks on each page accessed, and consequently lock management.



High Access Small Table Allocation

Figure 9

9. Summary

HP SQL is HP's relational database management system. HP SQL was designed to support a high level of concurrency and still maintain data integrity. As the knowledge of the concurrency mechanism in HP SQL increases, the throughput of applications developed can be maximized. To achieve this goal, the functionality and design of locking were discussed. The different locks involved in SQL operations were described. Compatibility as well as conversion rules and their effects were studied. The design of the lock manager, and the processing of lock requests were presented. The locking activities on all objects in a database were overviewed for the different SQL commands. Finally, some performance notes were summarized. This paper has hopefully presented clearly the functionality and design of concurrency control in HP SQL. HP SQL is currently available on HP-UX, MPE/XL, and MPE/V operating systems.

Acknowledgments

The author wish to thank Scott Walecka for his prompt and invaluable multiple reviews of this paper. The author would also like to thank Mary Loomis, Emmanuel Onuegbe, Frank Dean, Dave Wilde, and Alex Carlton for providing numerous comments to improve the presentation of this paper.

HP SQL Performance

Edward C. Cheng
Distributed Data Management Laboratory

Hewlett-Packard
19447 Pruneridge Avenue
Cupertino, CA 95014

ABSTRACT

A sophisticated database management system (DBMS) should provide more than just the classical data management in a consistent state. It should have the intelligence to determine the most efficient way to manipulate data and should allow sufficient flexibility for users to control concurrency and thereby, to control performance.

HP SQL is a relational DBMS which permits users to define and access relational data objects. By default, HP SQL runs on the highest consistency level to maintain data integrity and at the same time provides maximum possible concurrency. However, some users may wish to make tradeoffs between the most stringent consistency requirements and a need for higher concurrency. HP SQL therefore also furnishes users with channels to communicate with the system about the type of operations they would perform within a transaction so that the system can utilize the appropriate lock modes to ensure data integrity with the highest possible concurrency.

This paper highlights the key factors used by HP SQL in choosing the optimized access path for a query. It also describes how a user can affect the optimizer in making such choices as well as influence the consistency level that a transaction employs. The result of some recent benchmark runs will be used to illustrate the benefits of features discussed in the paper.

Key Words and Phrases: Query Optimization, concurrency, selectivity factor, isolation levels, relational databases, DBEfileset, SQL

1. Introduction

When database users from the early 70's switched from the hierarchical and network models to relational technology, they appreciated the flexibility of relational databases but were also disappointed because of the typical relative degradation in performance. In a relational database, information is logically arranged in rows and columns which are defined in tables or relations. Tables are related to each other by key columns. A user can therefore manipulate information through retrievals and join operations on these logical units without specifying in what way the access is to be done. The query optimizer in the database management system (DBMS) is responsible for deciding the path for accessing the data. In other words, the performance of the DBMS relies heavily upon the ability of the query optimizer in choosing an optimal access path for an issued query. Moreover, in a multi-user environment, the throughput performance of the database system also depends on how much concurrency is allowed by the system.

In this paper, I will first look at the criteria that the query optimizer of HP SQL uses in choosing access paths. Second, I will disclose the internal locking scheme of HP SQL and how it uses this algorithm to enforce data integrity and at the same time provide maximum possible concurrency. The paper will then focus at the various channels that HP SQL provides to users to control the degree of concurrency, resulting in significantly higher throughput performance under certain application environments. Finally, I will conclude the discussion by showing the result of some benchmark runs.

2. Query Optimization

To facilitate the discussion, I will use a series of example queries against a database. The following tables are assumed to be in the database:

```
CUSTOMERS (CUSTNO, NAME, BRANCHNO, BALANCE)
SALESPERSON (SALESPERNO, REGIONNO, SALETODATE)
SALESREGION (REGIONNO, SALETODATE, LOCATION)
```

I also assume indices exist on these columns:

```
CUSTOMERS.CUSTNO, CUSTOMERS.REGIONNO,
SALESPERSON.SALESPERNO, SALESPERSON.REGIONNO,
SALESREGION.REGIONNO.
```

Consider the following set of queries:

Q1: Find the balance of the account whose account number is 10005.

```
SELECT BALANCE
FROM CUSTOMERS
WHERE CUSTNO = 10005;
```

Q2: Find the account numbers of all customers named "John Smith".

```
SELECT CUSTNO
FROM CUSTOMERS
WHERE NAME = 'JOHN SMITH';
```

Q3: Find the customers who have a balance of between 5,000 to 10,000.

```
SELECT CUSTNO, NAME
FROM CUSTOMERS
WHERE BALANCE > 5000 AND BALANCE < 10000;
```

Note that each of these queries accesses only one table. They are therefore refered to as single-relation queries. Optimization of these queries simply means the selection of an optimal path to access the relation. By contrast, the following query is a join query over multiple tables:

Q4: Find the location of the teller whose teller number is 150.

```
SELECT LOCATION
FROM SALESPERSON, SALESREGION
WHERE SALESPERNO = 150
AND SALESPERSON.REGIONNO = SALESREGION.REGIONNO;
```

To obtain the result of this query, the DBMS has to access both the SALESPERSON and SALESREGION tables. The optimizer therefore has the responsibility of looking for the best join permutation to do the job, the best join method to be used, and finally the best access path for retrieving data from each table.

3. Single Relation Query Optimization

As pointed out above, optimization of a single-relation query is simply choosing the cheapest access method for that query on the relation. HP SQL accomplishes this task by comparing the costs of all possible access paths. This is done when the query is preprocessed.

In the first stage of preprocessing, all internal information about the table in question is brought into memory. This includes information describing the table itself, the indices that are built on this table, and the DBEfileset with which this table is associated. The cost of a relation scan is then computed by adding the number of pages in this table and the number of *page table pages* in the DBEfileset. The page table pages are used by HP SQL to describe the data pages in the DBEfileset. A sequential scan of the table requires scanning the page table pages in order to locate the table's data pages.

After the table scan method is evaluated, all possible index scans are considered. The cost of an index scan is calculated by adding the number of B-Tree pages and data pages that the system needs to visit. Unlike a table scan, this number of pages is governed by the *selectivity factor* indicated in the WHERE clause of the query. The selectivity factor is defined as the ratio of the estimated number of tuples satisfying the query over the total number of tuples in the relation. The costs of all index scans are compared and the cheapest one is then compared to the table scan cost to obtain the best plan.

For example, consider query Q1. The query optimizer will first evaluate the cost of scanning table CUSTOMERS sequentially. Second, both index access methods of using indices on CUSTOMERS.CUSTNO and CUSTOMERS.REGIONNO will be considered. Note that although only one data page will be visited if the index on CUSTNO is picked (assuming CUSTNO is a unique key), still the other index could produce a cheaper path since the cost also depends on how many B-Tree pages are fetched. Finally, the cheaper index scan cost is compared to the table scan cost and the best plan is selected.

In Q2, since the WHERE clause does not specify any indexed column, the query optimizer will assume the worst case of having to touch every data page even for index scans.

In Q3, notice that the WHERE clause has an AND logical operator over the two predicates of BALANCE. This will lessen the selectivity factor of retrieving data through the index on BALANCE, and the optimizer will take that into account in computing the cost.

4. Join Query Optimization

To handle join queries, in addition to evaluating different paths to access individual tables, the optimizer also has to decide on a join order and a join method for each join. In other words, the final solution of a join query will contain a join order over the tables, a join method for each join over a set of tables, and a plan to access each table.

If a query is joining N relations, then there are N factorial possible join orders. However, it is meaningless to consider orders that join tables which do not have join predicates between them. By looking at the WHERE clause, the optimizer first eliminates the meaningless join permutations. Next for each join order, a join method is chosen for each join presented by that join order. Currently HP SQL uses a nested loop join with modified scan.* In its upcoming release, HP SQL will also employ sort merge join. Note that for either join method, the join can be done over multiple tables at once, provided all join columns belong to the same order equivalence class [2]. For example, if the join predicates are:

T1.C1 = T2.C2 and T2.C2 = T3.C3

then a 3-way join can be done on T1, T2, and T3. Hence, by knowing the cost of accessing individual tables, the cost of each join is computed, and in turn, the cost of each join order is found. The cheapest plan is then picked as the final solution.

Also note that in considering access paths of a relation in a join query, all paths that return the interesting orders of that query are considered. Access paths are said to return interesting orders if they either present the joined columns or the ORDER BY/GROUP BY columns in order. Note that the access method which returns an interesting order does not require sorting of the table and that is why access paths which return interesting orders will always be considered.

Consider query Q4 as an example. The cheapest way and the ways that return the interesting orders of accessing SALESPERSON and SALESREGION are first found. Here, indices on SALESPERNO and REGIONNO are the interesting orders of the query since both of them are involved in the join predicate. No join order is eliminated; both SALESPERSON-SALESREGION and SALESREGION-SALESPERSON are considered. Both sort-merge and nested-loop-join methods are evaluated for each of these two pairs. The best plan is chosen by comparing the costs of join methods over the two join pairs.

Although HP SQL is responsible for query optimization, it is clear that the more accurate the information about the tables the optimizer can obtain, the better the decision it can

---

* Modified scan is a scan method to speed up a nested loop join by taking advantage of the memory buffer cache.

make in the selection process for both single-relation queries and join queries. It is therefore very important for the DBA to update the statistics of the relations after heavy loading, inserting, or deleting of data. The SQL command to do this is,

```
UPDATE STATISTICS FOR TABLE tablename;
```

This can be done either interactively or through a preprocessed program. In any case, it is advised to COMMIT WORK right after this command since update statistics would have to hold locks on a lot of the common resources in the database and thus affect concurrency.

Also note that since a table scan required a search through the page table pages in the DBEfileset, it is more efficient to define large relations in separate DBEfilesets. Small tables can be grouped together in the same DBEfileset without affecting the performance of a table scan, since each page table page can store information for up to 253 data pages.

In addition to influencing the optimizer, a user can also improve performance by telling HP SQL what kind of database functions are about to be performed. With such information, HP SQL can allow the maximum possible concurrency while data integrity is maintained. Let us first look at the locking scheme that HP SQL employs to contol data integrity and concurrency.

5. Basic Locking Algorithm of HP SQL

In HP SQL, all the five lock modes mentioned in [2] are implemented. They are identified as Share (S) to read, eXclusive (X) to update, Intention-Share (IS) and Intention-eXclusive (IX) to declare the intention to read and to write respectively, and finally Share-and-Intent-eXclusive (SIX) for reading the data and declaring the intention to update. When a user creates a table with the CREATE TABLE command, the table mode defaults to PRIVATE. With PRIVATE table mode, only one user can access the table at one time. Despite the type of application in process, the table is locked in exclusive mode. With PRIVATE table mode, although the lock manager in the DBMS does not have to deal with a complicated locking mechanism, no concurrency is allowed with this table.

To allow multiple users to access a table at one time, one can create the table and specify the table mode to be PUBLICREAD or PUBLIC. PUBLICREAD table permits concurrent users to read the table but at any one time only a single user can update the table. For read applications, intention share lock is applied to the table and share lock is granted to individual pages while exclusive table lock is used for update. For PUBLIC tables, the highest degree of concurrency is selected; intention locks are used as much as possible on the table level. The drawback is that now the system has more locking overhead. The rest of my discussion assumes the table is created in PUBLIC mode.

By default, HP SQL uses an implicit two-phase locking [1] strategy in order to guarantee transaction atomicity and serializability while multiple users are accessing a database. "Two-phase" here simply means that locks are issued as data objects are touched (first phase) and released at commit time (second phase). We also use a three-level locking hierarchy (relation, page, and tuple) with intention locks in order to increase the performance of detecting locking conflicts. Deadlock checking is done based on transactions and is done when a lock request has to wait. Table 1 shows the lock modes corresponding to different operations with the associated access paths. The access path is decided by the query optimizer described in the preceding section.

			Select	Update/Intent Update
Access Methods	Relation Scan	Table	S	SIX
		Page	-	SIX
	Indexed Scan	Table	IS	IX
		Page	S	S: Non-leaf; SIX: Leaf & Data pages

Table 1 Locking Strategy of HP SQL

Note that HP SQL has the ability to interpret the intention of a user in doing update and therefore a share and intent update (SIX) lock is granted when the following SQL commands are issued:

```
DECLARE CURSOR ... FOR UPDATE;
FETCH C1;
```

Now when the user asks for an update with cursor, no promotion of lock mode is required. With this intent update locking scheme, we have eliminated deadlock by 100% in test program U1A (see below).

Although applications running under this environment can guarantee a repeatable read consistency state [1,3], it is desirable for some transactions to be run under a less severe isolation level so that a higher degree of concurrency is observed in the system and better throughput results.

6. More Concurrency with Cursor Stability

HP SQL allows a user to specify for a transaction its isolation level. A higher degree of isolation means less concurrency in the database environment but ensures all data touched by a user to be consistent throughout the transaction. By default, all transactions are run on a high level of isolation to maintain repeatable read. Some applications require this level of control, since within the transaction a user may want to repeatedly read a data object in a consistent state. However, for those applications which either do not have such a strict requirement or do not need to revisit certain data objects within the same transaction, a lower isolation level and consequently more concurrency would be advantageous.

It is for this reason that HP SQL furnishes the syntax to specify a lower isolation level called cursor stability (CS), also known as non-repeatable read. A transaction running on CS level will only hold locks on pages in one of the two categories:

- update has been done to the pages

- the cursor is currently scanning the pages

A cursor here is an internal pointer of the DBMS used to scan data. Note that it is necessary to hold all exclusive locks until commit work to ensure that no uncommitted data can be read by other transactions. With this feature, the DBMS can allow more transactions to run concurrently in the database, but the disadvantage is that the transaction might find inconsistent data if it went back to those pages it has read. Users of this feature should be aware of this impact; applications which expect repeatable-read characteristics should not be run on this isolation level. The following is the syntax to specify the CS isolation level for a transaction:

```
BEGIN WORK CS;
```

Table 2 shows the lock modes of different operations under cursor stability.

			Select	Update/Intent Update
Access Methods	Relation Scan	Table	IS	IX
		Page	S*	SIX*
	Indexed Scan	Table	IS	IX
		Page	S*	S: Non-leaf; SIX*: Leaf & Data pages

*: Release lock on the next fetch

Table 2 Lock Modes with Cursor Stability

Using CS in benchmark test U2 shows a tremendous amount of improvement in throughput performance. Other isolation levels are under investigation and may be implemented for future releases of HP SQL [4].

7. Further Improvement with DML-Only Mode

When an application program is preprocessed, an optimized plan is generated for each query in the program. This plan is compressed and stored in the database. In HP SQL, an access plan is also known as a section. At any later time when the query is executed, the DBMS will pull out the corresponding section from the database and execute it.

In general, sections in the same transaction are linked up together as a list and kept in memory until the end of the transaction so that re-execution of a query does not require setting up the corresponding section again. At commit time, this section list must be purged since a plan can become invalid and cause a repreprocess to occur if the access paths specified in the plan are removed (e.g. an index is dropped) or the user who ran the program lost his authority in accessing part or all of the data involved in the plan (e.g. his select privilege on a table is revoked).

A section can only be invalidated by either Data Definition Language (DDL) or Data Control Language (DCL). Because the bulk of today's database applications deal with only Data Manipulation Language (DML) (rather than DDL or DCL), it is useful to offer a way to declare that only DML is to be issued in the environment. From that point on, preprocessed queries (i.e. sections) will be read from the database when the query is first executed and will then stay in user's local memory as long as the user is connected to the database. This eliminates the CPU time required to re-fetch the stored plans from the database and to undo the compression of the plans. This feature is called "section caching across transactions".

To trigger this option in HP SQL, a user only needs to disable DDL commands through a utility provided with the DBMS, namely SQLUtil. An improvement of over 25% in throughput is observed with this feature.

8. Some Experimental Results

Two sets of test programs were used to illustrate the performance impact of the above issues. The first set is the U1A and U1B programs. Each transaction in these programs is doing update operations (simulating sales operations in a business environment) on the three tables described in section 2. The queries are shown below:

```
UPDATE CUSTOMERS
SET BALANCE = BALANCE + :AMOUNT
WHERE CUSTNO = :CUSTNO;

UPDATE SALESPERSON
SET SALETODATE = SALETODATE + :AMOUNT
WHERE SALESPERNO = :SALESPERNO;

UPDATE SALESREGION
SET SALETODATE = SALETODATE + :AMOUNT
WHERE REGIONNO = :REGIONNO;
```

The second benchmark program we used is U2. There are three basic differences between U1 and U2 programs. First, for each update transaction in U2, a unique voucher number is assigned. This voucher number is serialized by using a ORDERNO table. The ORDERNO table is a single-tuple relation. Second, beside doing update, some transactions in U2 also do select operations (to simulate inquiries in a sales office). Third, there is an option in U2 which allows simulation of report writers or bookkeeping types of applications in the background. This is one sample background query we used:

```
BULK SELECT CUSTNO, NAME, BALANCE
INTO :BUFFER
FROM CUSTOMERS
WHERE BALANCE BETWEEN :LIMIT1 AND :LIMIT2;
```

The sizes of the tables are shown below, for U1 and U2 respectively:

Table 3 Sizes of Relations

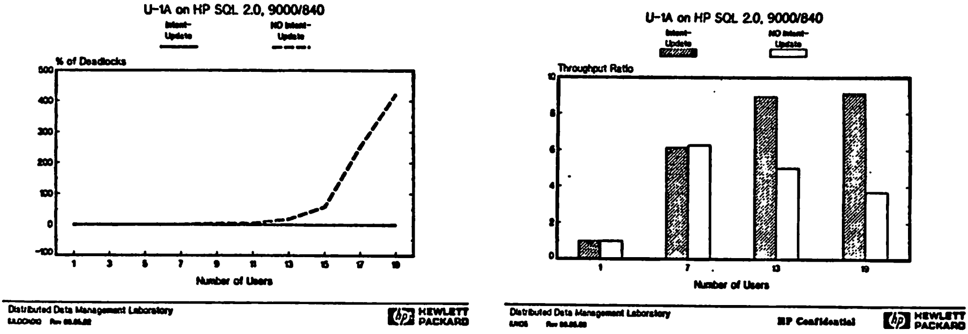
	No. of Rows	No. of Bytes per Row
CUSTOMERS	2,000,000	96
SALESPERSON	2,000	96
SALESREGION	200	96

(a) U1

	No. of Rows	No. of Bytes per Row
CUSTOMERS	500,00	192
SALESPERSON	1,000	168
SALESREGION	100	192
ORDERNO	1	4

(b) U2

The following figures show the effects of the enhancements described above. Figure 1 shows the impact of intent update locking in U1A. 1(a) shows the increase in the number of deadlocks when more users are added to the system. With intent update lock, however, the number of deadlocks is reduced to 0. Figure 1(b) shows throughput with and without intent update locks. Note that when the number of deadlocks increases, the new locking scheme becomes more significant.



(a) (b)

Fig. 1. Improvement with Intent Update Locking Algorithm

Figure 2 shows the influence of cursor stability on U2 with the background process. The background job is doing a table scan over the CUSTOMERS table. By default, this would lock any update transaction out for the entire period of its operation. Cursor stability allows the background process to release locks along the way, the benefit is outstanding. This is run with all tables in one DBEfileset.

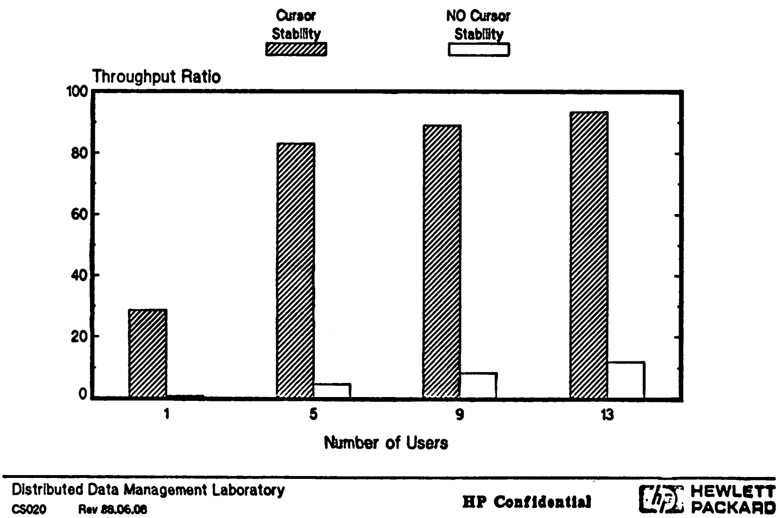
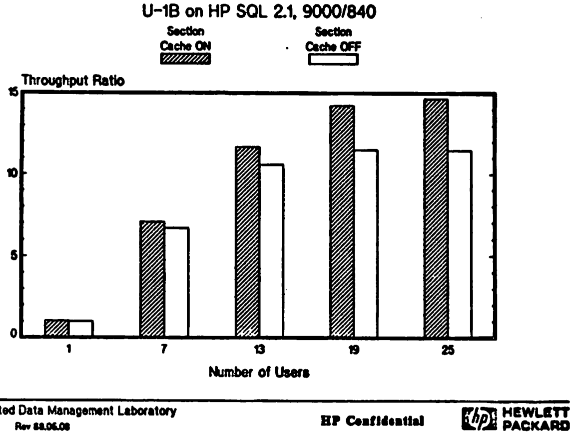


Fig. 2. Cursor Stability on U2

Finally, figure 3 shows the power of section caching across transactions in both the U1B and U2 benchmark tests. The gain in throughput is over 25% in both cases. The reason that the percentage gain on U2 is higher than that on U1B is due to the fact that there are more sections in the U2 program.



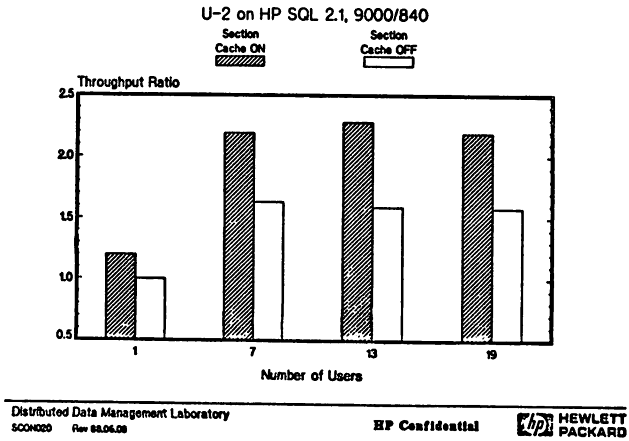


Fig. 3. Impact of Section Caching on U1B and U2

9. Summary

The focus of this paper is in the presentation of some of the schemes that we can contribute to a high performance database management system. The algorithms presented have all been implemented in HP SQL. A main motivation for these performance enhancements is the fact that we believe it is very important to have high performance on single-user, single-query applications, as well as multiple-user environments.

Acknowledgements

I would like to thank Vish Krishnan, Tony Marriott, David Ai, and Alberto Lutgardo, for supporting me in the design and implementation of the performance enhancement project. The inspiration and support from Jay Veazey, Reza Taheri, and Henry Cate are very much appreciated. My friend and colleague, Birgit Luebke has given me a lot of suggestions and encouragement in benchmarking HP SQL, thanks. Ragaa Ishak has provided both technical and editing comments that have greatly improved the presentation of this paper. Also thanks to May Kovalick who has helped in making this project possible.

Reference

1. J. N. Gray, Notes on Data Base Operating Systems, IBM Research Laboratory San Jose, CA. 1977

2. P. Griffiths Selinger, Access Path Selection in a Relational Database Management System, IBM Research Laboratory, San Jose, CA. 1979

3. R. K. Ishak, Concurrency Control in HP SQL, Distributed Data Management Laboratory, Cupertino, CA. 1988

4. A. Lutgardo, The HP SQL Advance - Towards the OLTP Market, Distributed Data Management Laboratory, Cupertino, CA. 1988

HEWLETT PACKARD

PC NETWORKING ALTERNATIVES

PREPARED BY

CHRIS OLSON

HEWLETT-PACKARD COMPANY
NEELY SALES REGION
3003 Scott Boulevard
Santa Clara, California 95054

MAY 1988

TABLE OF CONTENTS

HEWLETT PACKARD PC NETWORKING ALTERNATIVES

Hewlett-Packard offers a variety of PC networking products that allow PCs to be connected to one another as well as to an HP 3000. In a PC network, either a PC or an HP 3000 can be used as a server. The benefit of a network is that users can share data and peripherals. This sharing results in cost savings and/or organizational efficiencies.

The four options presented here assume that an HP 3000 is already installed and operating in an existing data processing environment. These options build upon this environment and further utilize the multi-tasking capabilities of the HP 3000 computer. These networking strategies represent those configurations most frequently used in an office environment and address a multitude of PC networking needs.

Options #1, #2, and #3 basically offer the same capabilities; shared disc, shared printers, shared files, and the ability to perform tape back-ups of all disc drives. Option #4, which is not a network, offers shared printers and shared files. All four options vary in price, performance (speed), and capabilities. In order to compare these strategies, a sample configuration is presented for each option which assumes that two departments or work groups are implementing a PC network, and that each group of PCs has access to an HP 3000 Series 70.

While these strategies are described separately, the final networking configuration for a particular organization may be a combination of one or more of these options. In order to determine which option is best, additional information will need to be gathered. This information includes disc and print volumes, tasks performed most frequently (e.g. word processing, graphics etc.), physical dimensions of the building(s), expected growth, budget, and the importance of speed, versus functionality, versus support.

None of these options address specific software needs. Again, more information will need to be gathered to recommend specific packages. However, each option includes Hewlett-Packard's most popular PC application bundles, the Vectra Office Professional. This bundle includes an easy to use word processor, graphics, a database application, a spreadsheet application, and a terminal emulator.

It is important that this paper be used only as a guideline in determining a PC networking strategy due to the many issues surrounding the implementation of PC networks. Because Hewlett-Packard is committed to the long-term success of each of its customers, it is strongly recommended that customers considering the implementation of a PC network work closely with the Hewlett Packard support team to ensure maximum success.

OPTION #1: ADD-ON MICRO 3000GX AS A SERVER

(Business System 3000 with Business System Plus Software)

The Business System 3000 (MICRO 3000GX) is a complete entry level integrated business system which is ideal for departmental applications where local area networking and PC integration is desired. In this configuration, the HP MICRO 3000 minicomputer acts as a server to PCs which are connected via twisted pair type cabling (HP StarLAN). PC users have access to all of the MICRO 3000GX's peripherals including printers, discs, plotters, and tape drives. Users can have private and shared PC files on the MICRO 3000GX, eliminating the need for floppy diskettes to share and store data. Additionally, PC hard discs can be backed up to the HP 3000, significantly reducing the risk of lost PC data. Also, by connecting the Business System 3000 to the existing HP 3000, all users on the network will have access to applications on the HP 3000 such as HPDeskManager.

Business System Plus software is a key component of the Business System 3000. In the sample configuration, Business System Plus includes:

1) Software to allow PC users access to the MICRO 3000GX peripherals;

2) Utilities that aid in the configuration and maintenance of the network, and;

3) Five key PC applications for each PC on the network. The five applications are Advancelink, Executive Card Manager., Executive Memomaker, Graphics Gallery, and Lotus(R) 1-2-3(R).

As an option, electronic mail (HPDeskManager) and software that allows a PC access to Image databases and the ability to download data to the PC (Information Access) can be added to Business System Plus. These two components are not included in the pricing of the sample configuration.

The following is a description of the advantages and disadvantages of this option versus the other options being presented.

<u>Advantages</u>

- o Ability to print/plot to more than one printer/plotter at the same time -- the HP 3000 uses a multi-tasking operating system (MPE) which can manage several devices at a time.

- o Excellent spooler manipulation capabilities via the MPE spooler (SPOOK), -- capabilities include the ability to suspend printing, re-start printing without loss of data, alter print priorities and the number of copies to be printed.

- o Offloads existing HP 3000 by running on a completely separate HP 3000.

- o High security due to built in MPE security.

- o Ability to use the same printer for both PC applications and HP 3000 applications such as HPDesk.

- o With Business System Plus, new versions of PC applications can be downloaded from the HP 3000 to all PCs on the network. This greatly reduces installation time and aids in "version" control.

- o Provides centralized control -- multiple servers do not need to be maintained. The MICRO 3000GX could be located in the computer room for easy maintenance by the MIS staff.

- o Already have personnel trained on the HP 3000. The only additional training would be on the use of the network software. Backups and routine maintenance would be done in the same manner as the existing HP 3000.

- o Remote PC users can access the network via HP SERIAL.

Disadvantages

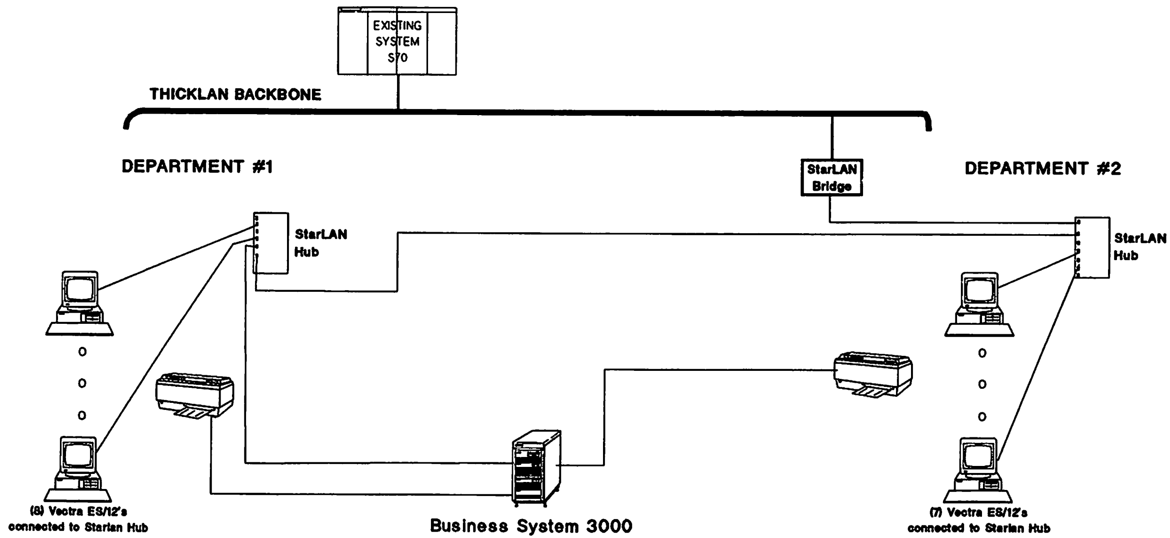
- o Costs more than a PC server due to its increased functionality.

- o Performance of transaction based PC network software such as database applications will perform noticeably slower on an HP 3000 server than they would on a PC server. This is due to the extra overhead associated with running the network over MPE.

- o If HP 3000 has a system failure the network will go down. Users can continue to work on their PCs in local mode but will be unable to print or get at files on shared discs attached to the HP 3000.

OPTION #1
ADD-ON MICRO 3000 AS A SERVER

Business System 3000 with Business System Plus Software



HEWLETT
PACKARD

OPTION #1: MICRO 3000GX AS A SERVER
(Business System 3000 with Business System Plus Software)

CONFIGURATION AND PRICING FOR 15 USERS

			(List Prices)
32519A	Business System 3000 - includes MICRO 3000GX, 4Mb main memory, 304 Mb disc drive, 67 Mb embedded cartridge tape, 8 ports (4 with modem capability), 700/92 console with cable, StarLAN Link, and StarLAN Hub		27,140
32508A	Business System 3000 Software - includes HPEasytime, Disc Caching, and LAN Link 3000 Software		0
51450A	FOS - Fundamental Operating System		0
OPT. 001	Integrated System Shipments		0
33440A	Laserjet Series II Printer with RS-232 and Centronics interface	2x 2,595	5,190
13242X	25-pin M/3-pin M RS-232 serial cable, 5m	2x 80	160
27212A	HP StarLAN Hub	2x 1,415	2,830
28647A	HP StarLAN-10Mbps 802.3 Bridge		5,000
OPT. 903	US/CAN 125V Power Cord		
32510A	Business System Plus for MICRO 3000GX includes Resource Sharing, Network utilities and 10 right to copy for Lotus(R) 1-2-3(R), Exec. Memomaker, Exec. Card Mgr., Advancelink, and Graphics Gallery		29,220
OPT. 311	Deletes Information Access		- 4,250
OPT. 312	Deletes HPDeskManager and AdvanceMail		- 6,380
35510L	Business System Plus License Upgrade for 10 additional PCs		21,595
OPT. 311	Deletes Information Access		- 4,030
OPT. 312	Deletes HPDeskManager and AdvanceMail		- 2,730
OPT. 313	Deletes 10 right to copy for Lotus(R) 1-2-3(R), Exec. Memomaker, Exec. Card Mgr., Advancelink, and Graphics Gallery		- 13,865

68300F	Vectra Office Software Pack includes Lotus(R) 1-2-3(R), Exec. Memomaker, Exec. Card Mgr., Advancelink, Graphics Gallery, and AdvanceMail	5x	1,650	8,250
50923F	Network Services-User Services For HP Vectra and IBM PCs			100
OPT. 0A9	License for 10 users			800
50923F	Network Services-User Services For HP Vectra and IBM PCs	5x	100	500
50926F	StarLAN/HP Vectra PC Link	15x	395	5,925
50929F	LAN PC user configuration and diagnostics package for HP Vectra			90
OPT. 0A9	License for 10 users			720
50929F	LAN PC user configuration and diagnostics package for HP Vectra	5x	90	450
		Total		76,715

OPTION #2: PC'S AS SERVERS

(Vectra Servers with Officeshare Networking Software)

Hewlett Packard's local area network offering is actually a family of products called Officeshare. The family consists of three products: HP StarLAN, which is based on twisted pair cabling; HP THinLAN, which is based on co-axial cabling, and; HP SERIAL, which is based on RS 232 connection and is used for connecting remote PCs to the HP 3000 server. (HP SERIAL is not available on a PC server.)

In general, HP recommends HP StarLAN be used for the office environment because existing phone wire can often be utilized. With an HP Vectra PC server, StarLAN supports disc and file sharing, printer sharing with spooling, and plotter sharing with spooling. In order to implement two departments using PC Servers, HP would recommend that each department use its own local server. These servers would then be networked to each other, allowing users to share data and printers on both servers. In addition, both servers would be networked to the existing HP 3000 which would give users access to applications on the HP 3000 such as HPDeskManager.

The following is a description of the advantages and disadvantages of this option versus the other options being presented.

<u>Advantages</u>

- o Best overall performance, particularly for transaction based PC network software such as database applications.
- o Offloads the existing HP 3000.
- o Relatively low cost.

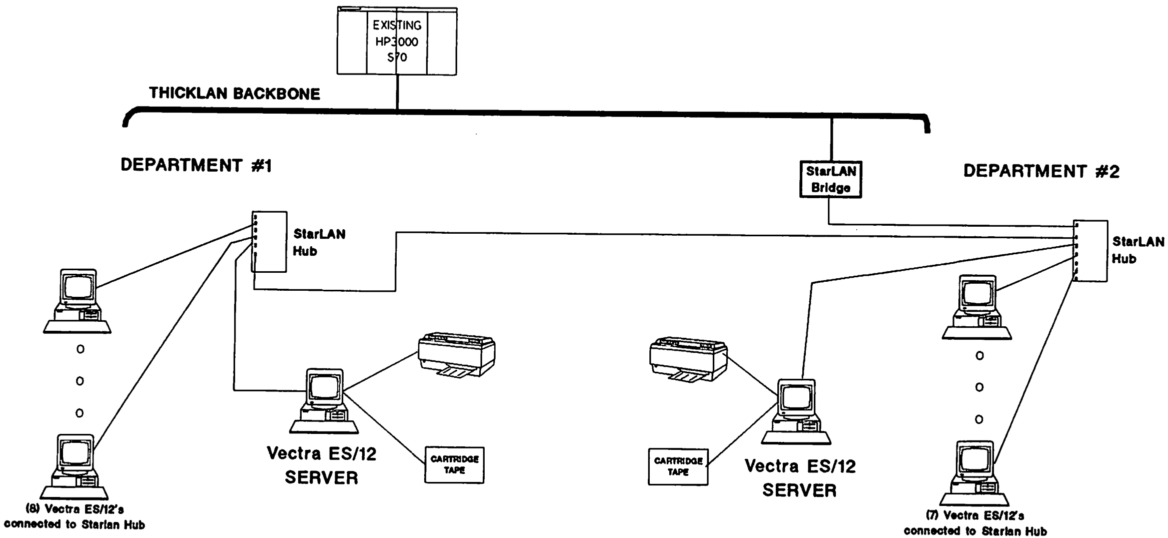
<u>Disadvantages</u>

- o Doesn't handle simultaneous printing and/or plotting due to MS-DOS being a single threaded operating system. Only 1 printer/plotter per server can be active at a time. This means if the plotter is active, nothing can be printed until it has finished. Conversely, if the printer is printing a large document, no other printing or plotting can take place.
- o Low security. Data on the network could be accessed. Having security violations could be minimized by placing the server in a secured location.
- o Someone must be trained and assigned to do routine network maintenance such as backups on each server.
- o Remote PCs cannot access the network.

OPTION #2
PC'S AS SERVERS

Vectra Servers with Officeshare Networking Software

2060



OPTION #2: PC'S AS SERVERS
(Vectra Servers with Officeshare Networking Software)

CONFIGURATION AND PRICING FOR 15 USERS

				(List Prices)
D1340A	Vectra ES/12 Model 40 includes one 5.25 1.2Mb flexible disc drive and a 40 M byte hard disc	2x	4195	$8,390
50925F	PC Server Software for the HP Vectra	2x	970	1,940
50926F	StarLAN/HP Vectra PC Link	2x	395	790
9144A	67M bytes (formatted) 1/4-inch Cartridge Tape Drive including Disc Tape interface and cable	2x	2,250	4,500
33440A	Laserjet Series II Printer with RS-232 and Centronics interface	2x	2,595	5,190
24542G	9-pin F/25-pin M serial printer cable 3M	2x	55	110
28647A	HP StarLAN-10Mbps 802.3 Bridge			5,000
OPT. 903	US/CAN 125V Power Cord			0
27212A	HP StarLAN Hub	2x	1,415	2,830
68300F	Vectra Office Software Pack includes Lotus(R) 1-2-3(R), Exec. Memomaker, Exec. Card Mgr., Advancelink, Graphics Gallery, and AdvanceMail	15x	1,650	24,750
50923F	Network Services-User Services For HP Vectra and IBM PCs			100
OPT. 0A9	License for 10 users			800
50923F	Network Services-User Services For HP Vectra and IBM PCs	5x	100	500
50926F	StarLAN/HP Vectra PC Link	15x	395	5,925
50929F	LAN PC user configuration and diagnostics package for HP Vectra			90
OPT. 0A9	License for 10 users			720
50929F	LAN PC user configuration and diagnostics package for HP Vectra	5x	90	450
		Total		$ 62,085

OPTION #3: EXISTING HP 3000 AS A SERVER

(Existing HP 3000 with Business System Plus Software)

Option #3 is basically the same as Option #1 except instead of using a Business System 3000 (MICRO 3000GX) as the server, the existing HP 3000 and Business System Plus software is used. A description of Business System Plus software is given under Option #1.

With this option, the existing HP 3000 can continue to run HP 3000 applications and also act as a server to PCs connected to it via twisted pair cabling (HP StarLAN). These users will have access to applications running on the HP 3000 such as HPDeskManager via this connection. PC users will also have access to all of the HP 3000's peripherals including printers, discs, plotters, and tape drives. Users can have private and shared PC files on the HP 3000, eliminating the need for floppy diskettes to share and store data. Additionally, PC hard discs can be backed up to the HP 3000, significantly reducing the risk of lost PC data.

As an option, electronic mail (HPDeskManager) and software that allows a PC access to Image databases and the ability to download data to the PC (Information Access) can be added to Business System Plus. These two components are not included in the pricing of the sample configuration.

The following is a description of the advantages and disadvantages of this option versus the other options being presented.

<u>Advantages</u>

- o Already own the server which is the existing HP 3000.
- o Backup of PC data is done as part of normal system backup since PC data is stored in groups and accounts on the HP 3000.
- o High security due to MPE security and because the server (HP 3000) is typically located in the computer room which is a controlled environment.
- o Can use the same printer for both PC applications and HP 3000 applications such as HPDeskManager.
- o Can print/plot to more than one printer/plotter at the same time, -- the HP 3000 uses a multi-tasking operating system (MPE) which can manage more than one device at a time.
- o Excellent spooler manipulation capabilities via the MPE spooler (SPOOK). These capabilities include the ability to suspend printing, re-start printing without loss of data, and alter print priorities and the number of copies to be printed.
- o With Business System Plus, new versions of PC applications can be downloaded from the HP 3000 to all PCs on the network. This greatly reduces installation time and aids in "version" control.

- Provides centralized control, -- multiple servers do not need to be maintained.

- Already have personnel trained on the HP 3000. The only additional training would be on the use of the network software. Backups and routine maintenance of the server (existing HP 3000) would be done in the same manner as is being done today.

- Remote PC users have access to the local area network via HP SERIAL.

Disadvantages

- Uses resources on the existing HP 3000 which may already be approaching maximum capacity.

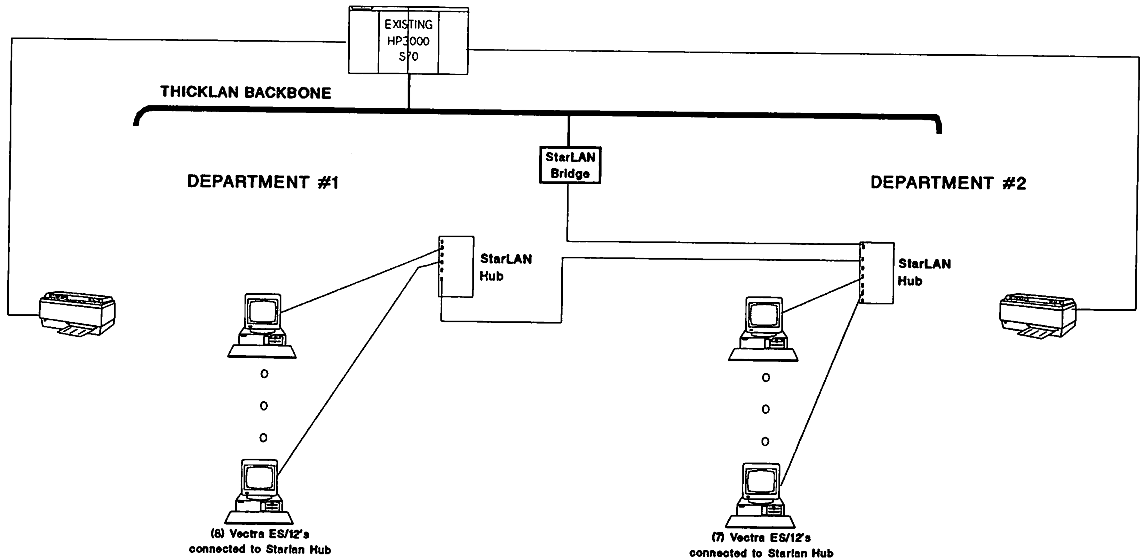
- Performance of transaction based PC network software such as database applications will perform noticeably slower on an HP 3000 server than they would on a PC server. This is due to the extra overhead associated with running the network over MPE.

- If HP 3000 has a system failure the network will go down. Users can continue to work on their PCs in local mode but will be unable to print, plot or get at files on shared discs attached to the HP 3000.

OPTION #3
EXISTING HP3000 AS A SERVER

Existing HP3000 with Business System Plus Software

2060



OPTION #3: EXISTING HP 3000 AS A SERVER
(Existing HP 3000 with Business System Plus Software)

CONFIGURATION AND PRICING FOR 15 USERS

			(List Prices)
32510C	Business System Plus for Series 68-70 includes Resource Sharing, Network utilities and 10 right to copy for Lotus(R) 1-2-3(R), Exec. Memomaker, Exec. Card Mgr., Advancelink, and Graphics Gallery		65,570
OPT. 331	Deletes Information Access		- 15,210
OPT. 332	Deletes HPDeskManager and AdvanceMail		- 23,295
35510L	Business System Plus License Upgrade for 10 additional PCs		21,595
OPT. 311	Deletes Information Access		- 4,030
OPT. 312	Deletes HPDeskManager and AdvanceMail		- 2,730
OPT. 313	Deletes 10 right to copy for Lotus(R) 1-2-3(R), Exec. Memomaker, Exec. Card Mgr., Advancelink, and Graphics Gallery		- 13,865
27212A	HP StarLAN Hub	2x 1,415	2,830
28647A	HP StarLAN-10Mbps 802.3 Bridge		5,000
OPT. 903	US/CAN 125V Power Cord		
68300F	Vectra Office Software Pack includes Lotus(R) 1-2-3(R), Exec. Memomaker, Exec. Card Mgr., Advancelink, Graphics Gallery, and AdvanceMail	5x 1,650	8,250
50923F	Network Services-User Services For HP Vectra and IBM PCs		100
OPT. 0A9	License for 10 users		800
50923F	Network Services-User Services For HP Vectra and IBM PCs	5x 100	500
50926F	StarLAN/HP Vectra PC Link	15x 395	5,925
50929F	LAN PC user configuration and diagnostics package for HP Vectra		90
OPT. 0A9	License for 10 users		720

50929F	LAN PC user configuration and diagnostics package for HP Vectra	5x	90	450
33440A	Laserjet Series II Printer with RS-232 and Centronics interface	2x	2,595	5,190
13242G	25-pin M/3-pin M RS-232 serial cable, 5m	2x	104	208
		Total		$ 58,098

OPTION #4: NON-LAN SOLUTION

(PCs with HPDeskManager and AdvancePrint)

HPDeskManager, AdvanceMail and AdvancePrint can be used to provide file sharing and printer/plotter sharing for PC users. File sharing can be accomplished by using AdvanceMail with HPDeskManager to distribute files to other users. Although this method is not as transparent as "true" file sharing, it can be used to accomplish the same task. Printer/plotter sharing can be accomplished by using AdvancePrint.

AdvanceMail is a PC application that provides electronic mail between the PC and HPDeskManager. AdvanceMail users can exchange messages and files, including any MS-DOS file, with any HPDeskManager or AdvanceMail users. A user wishing to share a word processing document, spreadsheet or any PC file could accomplish this by sending the file using AdvanceMail, which in turn would use HPDeskManager to transport the file to the user's HPDeskManager In-Tray.

AdvancePrint is an application that allows PC users to share printers and plotters which are attached to the HP 3000. These printers and plotters can be used for both PC applications as well as HP3000 applications. This eliminates the need to have separate printers/plotters, one for PC applications and one for HP 3000 applications.

The following is a description of the advantages and disadvantages of this option versus the other options being presented.

Advantages

- Relatively low cost solution. For the most part, cabling already exists and many users are already connected to the HP 3000.
- Files can be distributed among PC users as well as terminal based users.
- Best possible performance for PC applications because all file access is local to the PC.
- One printer can be used for printing from PC based applications and HP 3000 applications.
- The same printer can be used for both PC applications and HP 3000 applications such as HPDesk.

Disadvantages

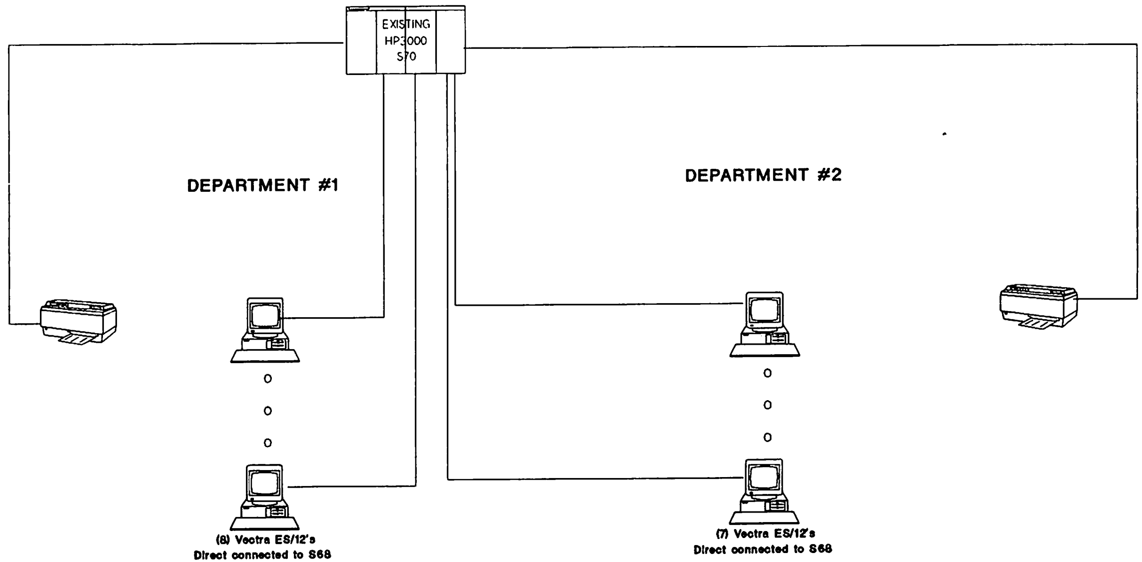
- No centralized storage of PC data. Data would have to be stored on each user's PC hard disc or on floppy diskettes.
- No transparent backup of PC data. Individuals would be responsible for backing up their own PC hard discs.

- o Cannot run networked PC applications. For example, if Lotus releases a networked version of Lotus(R) 1-2-3(R) that allows multiple users to use the same copy, this could not be run on the HP 3000 in its existing configuration. Networking software for the HP 3000 would have to be purchased in order to run any networked software.

OPTION #4
NON-LAN SOLUTION

(PCs with HPDeskmanager and AdvancePrint

2060



OPTION #4: NON-LAN SOLUTION
(PCs with HPDeskManager and AdvancePrint)

CONFIGURATION AND PRICING FOR 15 USERS

			(List Prices)
36570A	HPDeskManager		
OPT. 330	For use on a S70		25,000
32583L	AdvancePrint License for 10 users		1,530
32583F	AdvancePrint License for 1 user	5x 168	840
68300F	Vectra Office Software Pack includes Lotus(R) 1-2-3(R), Exec. Memomaker, Exec. Card Mgr., Advancelink, Graphics Gallery, and AdvanceMail	15x 1,650	24,750
33440A	Laserjet Series II Printer with RS-232 and Centronics interface	2x 2,595	5,190
13242G	25-pin M/3-pin M RS-232 serial cable, 5m	2x 104	208
		Total	$ 57,518

Fiber Optic Networking Update

Karen Dudley

Hewlett-Packard Company
Roseville Networks Division
8000 Foothills Boulevard
Roseville, CA 95678

Summary

Fiber optics is an exciting technology with great potential for networking. Declining prices and standards activity have helped increase market acceptance of fiber optics for data communications. This paper will describe the fiber optic LAN marketplace, fiber optic 802.3/Ethernet systems, Fiber Distributed Data Interface (FDDI), fiber optic MAP and HP Fiber Optic Link. The objective is to review fiber optic networking alternatives and HP's product strategy.

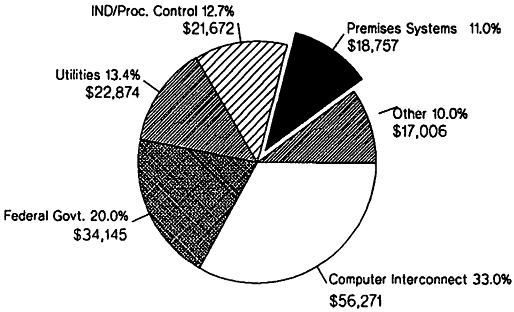
Today, fiber optics has been limited to specific applications that justify a premium price. The major network applications that require fiber today are campus backbones (distance and environmental requirements), secure environments, and harsh EMI/RFI environments such as the factory floor. Fiber is also justified for peripheral connections, especially mass storage, where large system configurability and data integrity are concerns. Very soon there will be increasing demand to use fiber for high performance workstation applications and larger capacity backbones.

The Fiber Optic LAN Marketplace

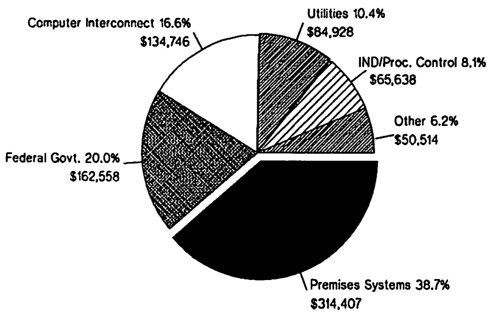
The fiber optic LAN market has not grown as quickly as industry analysts had initially predicted, but the current outlook is very promising. According to a Kessler Marketing Intelligence report, the fiber optic data communications market will grow from $171 million in 1987 to $813 million in 1992. The fastest growing application for fiber is premises systems, including local area networks, which is expected to rise from $19 million in 1987 to $314 million in 1992. Gartner Group expects steady growth in the fiber optic LAN market through 1988 and 1989, then accelerated growth as fiber based distribution systems become competitive with copper systems.

Fiber Optic Data Communications Market
Source: Kessler Marketing Intelligence

1987 Total = $171 Million



1992 Total = $813 Million



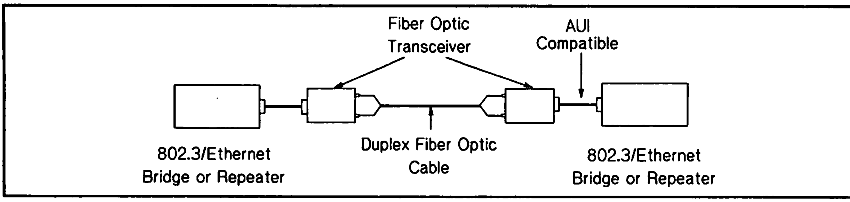
Fiber prices have dropped dramatically but high connect costs are still preventing wide-spread use of fiber optics in LAN applications. Fiber prices are now as low as 30 cents per meter depending on the fiber type, number of fibers and the quantity purchased. This is 30% lower than one year ago. Prices of other components (transceivers, connectors and electronics) have not declined as rapidly as the raw cable. Kessler Marketing Intelligence predicts further declines in fiber optic cable, transceivers and connectors by 1992.

Fiber optic LAN sales have been primarily to military and large financial institutions, where the security and reliability of fiber are worth the premium price. The key to widespread acceptance of fiber optic LANs will be standards development and continued decline in prices.

Fiber Optic 802.3/Ethernet Systems

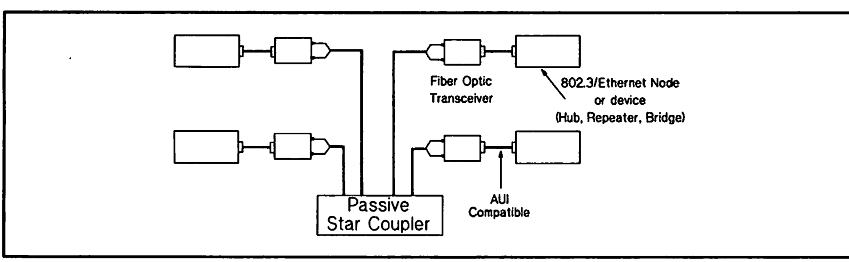
Fiber Optic 802.3/Ethernet Systems are now available from many vendors worldwide. Declining costs, simpler design and installation techniques, and standards activity are contributing to the growing acceptance of Fiber Optic Ethernet Systems. The cost for Fiber Optic Ethernet is $1,000 to $1,500 per node (approximately 25% more than a typical baseband LAN). The primary applications are for LAN backbones in harsh EMI/RFI environments and secure environments.

Fiber Optic Ethernet Systems available today include point-to-point links, passive star systems, active star systems and active ring systems. A point-to-point optical link can be configured using a pair of Ethernet optical transceivers with fiber optic cable between them. The transceivers can be connected to repeaters or bridges to create a Fiber Optic Inter-Repeater Link (FOIRL).

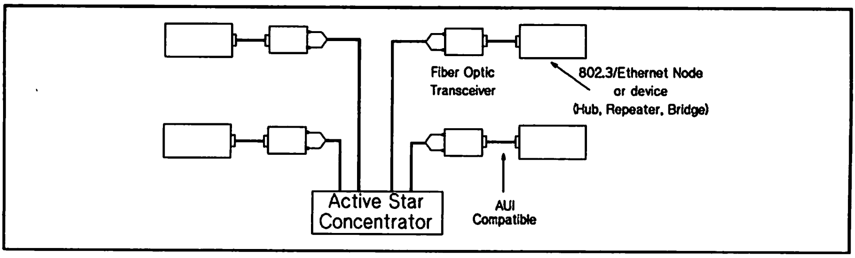


With a passive star 802.3/Ethernet system, a passive star coupler is used to provide the multiple access and broadcast functions of the CSMA/CD protocol used by 802.3/Ethernet. Fiber optic transceivers are used to connect 802.3/Ethernet nodes or devices such as multiport repeaters or bridges. The transceivers are connected to the star coupler with duplex fiber optic cable.

The use of a passive (requires no power) star coupler provides a highly reliable system where one node failure or a break in a cable effects only that node and not the entire system. The disadvantages of a passive system are imperfect collision detect and more complex design calculations (vs. an active star system). Passive systems are $300–$500 per node less than active systems.



Most active star systems consist of an Active Star concentrator with plug-in modules that connect optical transceivers. Active star systems provide better collision detection and simpler design calculations than passive star systems. Active Hubs typically include circuitry that can detect a failure in a node that would affect the network.



Another alternative for fiber optic Ethernet is an active ring configuration. Active ring systems provide a low cost implementation. Nodes are attached directly to a fiber optic ring with fiber transceivers. Adding nodes to a ring may require disconnecting the network. Node or cable failures may bring the ring down. Redundant rings can solve this problem but add cost. Rings are also more difficult to diagnose due to the lack of central electronics. When selecting a fiber optic 802.3/Ethernet System, be sure that it provides a fully compatible AUI connection to your nodes. Many are only partially compatible and this will cause connection problems in the future.

Standards Activity

The IEEE 802.3 committee has been addressing fiber optic LANs. A standard for a Fiber Optic Inter-Repeater Link (FOIRL) is complete. This standard defines the specifications for the fiber optic transceivers used in a point-to-point link.

Currently, there is an IEEE 802.3 working group evaluating fiber optic star standards. The committee is evaluating three proposals: passive star, asynchronous active star, and synchronous active star. The committee appears to be leaning towards a standard for passive stars and one for active stars. This standard should be complete by the end of 1988.

HP Product Strategy

Today HP references a Fiber Optic Transceiver for use with the HP 10Mbps to 10Mbps LAN Bridge and HP Repeater Kit. This solution provides a fiber optic Inter-Repeater Link to extend a baseband coax LAN between buildings.

In addition, HP is currently testing a Fiber Hub to provide a fiber optic 802.3 backbone to connect StarLAN, StarLAN 10 or ThinLAN subnets. Wiring recommendations from consultants such as Gartner Group are focusing on the use of fiber backbones with unshielded twisted-pair to the desk.

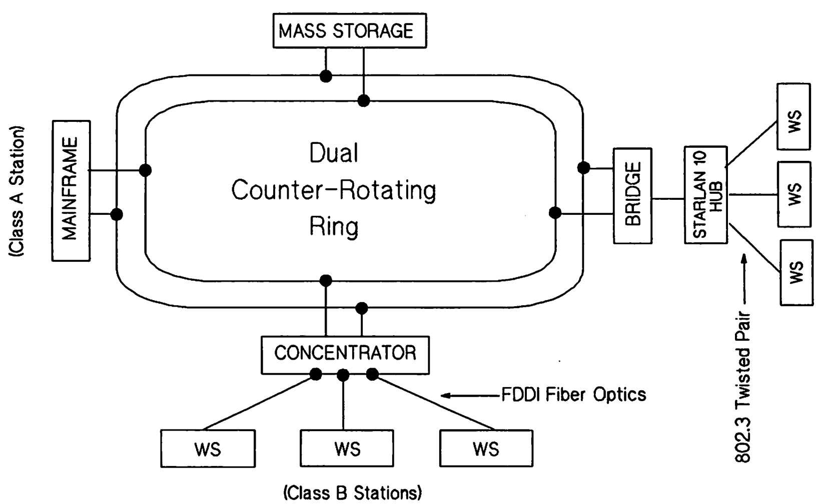
Fiber Distributed Data Interface

FDDI, being developed by the American National Standards Institute (ANSI) committee X3T9.5, is a counter-rotating token ring LAN with a data rate of 100Mbps. FDDI will support 500 dual attached stations linked by 100 km of duplex cable. A single station can support either a host computer or a subnetwork of hundreds of users.

The FDDI network consists of two independent 100Mbps rings – the primary and the secondary. The dual ring approach allows many different uses and configurations while providing redundancy and the ability to reconfigure the network under fault conditions. An FDDI ring has two basic station types: the class A station which connects to both the primary and secondary rings, and the class B station, which connects to either the primary or secondary ring but not to both. Class B stations require less hardware and are less expensive than class A stations and they can be easily isolated if a link fails. However, class A stations continue to operate in a reconfigured ring (under fault conditions).

FDDI also provides for hubs or wiring concentrators. With the use of hubs, the FDDI ring may be configured to look like a series of star networks or a branched tree network similar to a broadband network topology.

The FDDI Network

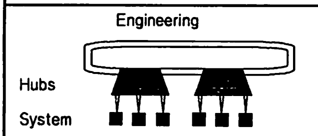
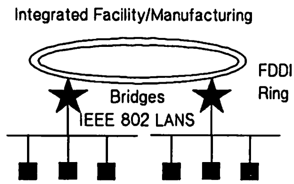
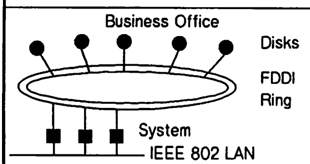


FDDI Applications

The original FDDI network was intended to be a high-performance backend network. Backend networks provide connections between computers and their high-speed peripherals or between computers in multiple processor environments. However, transport protocols and chips needed to efficiently support the long high-speed transfers typical of a backend network are still under development. Therefore, FDDI will first be used as an 802 network backbone, and to connect high-performance workstations.

FDDI will provide a high-performance backbone network to link together lower speed local area baseband networks (802.3, 802.4, 802.5) to support a greater number of stations and larger geographical distances. FDDI is ideal for this application due to its high performance, size, and compatibility with low performance standards such as 802.3 and 802.5. The FDDI MAC protocol provides a superset of the services required by the IEEE 802.2 logical link control. This simplifies the task of bridging FDDI to IEEE 802 networks. The initial FDDI specification is for data only networks and thus does not provide a complete multi-purpose backbone that is available with broadband today. New high-performance workstations and real-time imaging applications will drive the requirement for direct connection to FDDI LANs.

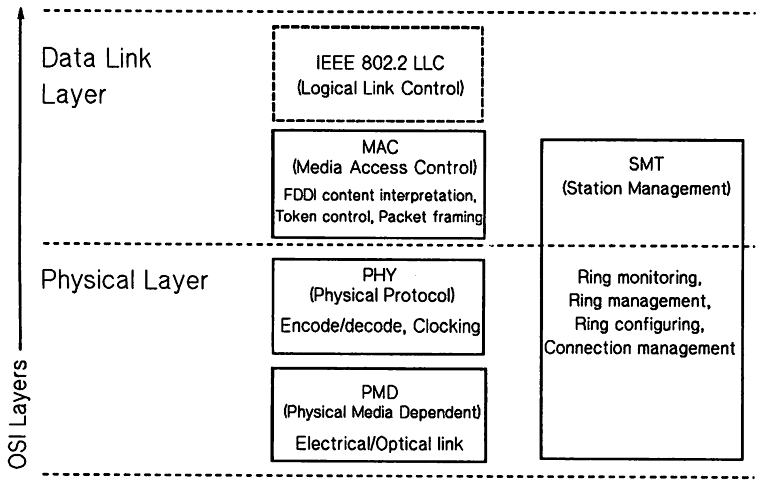
Customer Requirements

HP Solutions Fit	Type of Customer	Desired Topology	Customer Competitive Advantage
	Aerospace & Government & Universities	Workstation LAN	Performance
	Aerospace & Government & World-wide Manufacturers	Backbone	Integration
	Financial Institutions	Backend LAN	Security

Standards Activity

FDDI is being developed in accordance with the International Standard's Organization OSI model. FDDI represents the two lowest layers–Physical and Data Link–of the seven layer OSI framework.

FDDI Relationship to OSI Model



The PHY and MAC sections of the standard are complete. The PMD, which deals with optical connector standards is out for review. The SMT has the most technical work left to complete. HP is taking a very active role in contributing to the standard development. The initial FDDI standard is not expected to be complete until early 1989.

FDDI-II is a proposed follow-on to the original FDDI specification that adds circuit switching capability, thus expanding the applications of FDDI to include voice and video as well as data. FDDI-II uses a "slotted-ring" format. With this format, the 100Mbps capacity of the FDDI ring is divided into 16 channels. If FDDI-II works as envisioned, data, voice and some video signals could coexist in a single network. FDDI-II could solve many of the needs of factories, and provide a unified voice and data network for use in offices. FDDI-II standard work is still preliminary and a final standard is not expected for two to three years. A similar effort under IEEE 802.6 is also underway. It is expected that 802.6 will be positioned for public network applications and FDDI will be positioned for local area network applications.

FDDI Players

Over 50 vendors have supported FDDI standard development. Major system vendors include HP, Digital, IBM, Unisys, AT&T, Sun, and Apollo. Independent LAN vendors include Artel, Proteon, Fibercom, Fibronics, and NSC.

Fibronics, Fibercom, and NEC are the first vendors to announce FDDI products. Proteon and Artel have announced upgrades to current products when FDDI is available. Fibronics is the only vendor shipping a product today. Fibronics has a VME based system with a bridge to 802.3 Ethernet that sells for $36,800 per node. Prices are expected to drop to $25,000 per node once AMD integrated chips are available. Fibercom has announced an FDDI to Ethernet or token ring bridge that will be available mid 1989 and sell for $25,000. Integrated workstation products are expected late 1989 for approximately $10,000 per node.

HP Product Strategy

FDDI will play an important role for high bandwidth data backbones and high-performance workstation LANs. HP is contributing to the FDDI standard in the areas of fiber technology and networking. HP plans to use FDDI technology in future LAN products. Once the FDDI standard is complete, it will play an important role in multi-vendor networking.

Today HP offers a proprietary fiber optic connection for mass storage. HP Fiber Optic Link was developed specifically to solve problems inherent in using traditional copper media for large system configurations. HP Fiber Optic Link employs a simplified data transport protocol that is partially supported in custom VLSI, in order to maintain the data throughput required for backend communication. HP Fiber Optic Link will remain HP's fiber solution for the back-end until similar multivendor transport protocols and VLSI support circuits are developed for FDDI.

Fiber Optic MAP

The opportunity for fiber in factory applications exists where there are specific environmental problems such as nuclear power plants and process plants. The factory floor environment is susceptible to high electrical noise and interference. Fiber offers immunity to noise and complete electrical isolation of connected stations. Fiber will also provide a secure system for Aerospace and military equipment production.

There are a number of vendors producing fiber optic modems that conform to MAP standards but there is no real groundswell of support for fiber optic MAP today in the US. Contrasting the situation today in the US is the increasing support of fiber in the factory in Japan and Europe. Japanese manufacturers interviewed by Advanced Manufacturing Research indicated that they see the next manufacturing advance coming from the use of advanced communications to interconnect devices on the factory floor. To the Japanese, factory networking means working with fiber optics on the factory floor. A factory automation company has been formed by four companies (NTT, Tateishi Electric Machinery, Mitsubishi, Rayon and Sumitomo Denko) to develop fiber optic factory networks that conform to MAP).

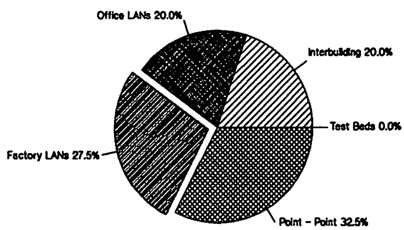
Standards Activity

A fiber optic MAP specification (802.4H) has been developed and is included in the appendix of the MAP 3.0 specification. This specification allows all topology options compatible with token passing bus Media Access Control layer. This includes linear passive bus, fiber optic passive star, fiber optic active bus, and fiber optic active star. The number of acceptable topologies included in this standard raises questions about the interoperability of different vendor's solutions.

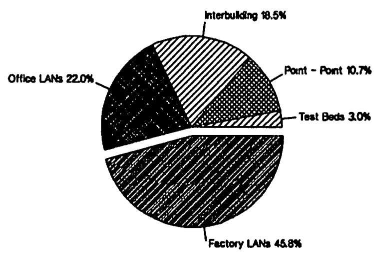
MAP/TOP Users Group Survey

A fiber optic survey was presented by Carl Morris (Chairman of the MAP/TOP Fiber Optics Special Interest Groups at the September 1988 MAP/TOP Users Group meeting. This was a survey of North American MAP/TOP Users Group. Approximately 1700 surveys were sent out and 195 were returned. Twenty percent of the survey respondents are using fiber today and 86% plan to use fiber in the future.

A. Today: 20% Using Fiber



B. Future: 86% Plan to use Fiber (Timeframe 88-90)



HP Product Strategy

HP is monitoring the 802.4H standards activity and the market for fiber optic MAP. The number of 802.4H topology options is a concern and HP will be evaluating the alternatives to help customers choose the best implementation. HP MAP products have been designed to accomodate Fiber Optic modems.

Wiring System Recommendations

Wiring systems are a very strategic hardware investment. The wiring system for a building or campus is very expensive and should last the life of the installation (10-20 years). Today, fiber optics is appropriate for specific applications that justify a premium price. While most end user applications today may not require fiber optics, you should consider running fiber for future use in new installations.

A cost effective strategy is to run "dark fiber" (currently inactive fiber) in campus or building backbones. This strategy provides bandwidth insurance at a reasonable cost. (See below for fiber optic cable specifications.) Fiber should not be run to the work area unless there is a specific requirement for security or applications that will require bandwidth greater than 10 Mbps. In general, multi-mode fiber optic backbones and unshielded twisted-pair horizontals will meet most users' requirements for the next 10 years.

Fiber Optic Specifications

The following fiber optic specifications will ensure that fiber optic pairs pulled today will support future standards such as FDDI.

- 62.5/125 micron fiber
- 500 MHz *km modal bandwidth
- cable attenuation ≤2.5 db/km

In general, HP recommends that ST connectors be used. FDDI will specify a new connector which is similar to the ST connector but provides the duplex connection required for FDDI.

Any Questions?
A Look at Questioning Techniques in the Classroom

Steven L. Mock
Hewlett-Packard Company
1421 S. Manhattan Avenue
Fullerton, California

The instructor finished the point he was trying to make, looked out at sixteen pairs of eyes and snapped "Any Questions?" Sixteen pairs of eyes stared back in silence as the instructor started his next point.

At one time or another, all of us involved in training have been in the above situation. As the instructor, we can only draw two conclusions from the silence: the students have a perfect understanding of everything we've said or they don't understand enough to ask any questions. In the interest of time and to keep on schedule, we move on.

In a training situation, too many times the burden of asking questions is placed on the student. This paper is an attempt to explore the various ways that we as instructors can place the burden on ourselves and ask questions instead of asking for questions. First, we will take a look at a commonly used five-step method of teaching. Next, we will discuss the types of questions an instructor can ask based on the answer we want to solicit. We will also discuss some questioning styles and techniques to use in various situations. Finally, we will see how questions can be used in each of the five steps of our teaching methods.

The goal of this paper is for you to discover how we as instructors can enhance the learning environment if we ask questions rather ask for questions.

THE FIVE STEP PROCESS

Motivation

Students come to training classes for many reasons. Some come to learn, others come to observe, still others come because they were told to. Whatever the reason, it is the instructor's job to motivate the student to learn. Different things motivate different people and sometimes it is a challenge to find what works.

Communication

Once we have motivated the students to learn, we must communicate the information to them. There are various methods of communication from which we can choose. They are as follows:

State It: This is the classic lecturing style of communication.

Use Examples: This method helps the student relate the learning to familiar situations.

Demonstrate it: This method lets the student actually see the learning in practice.

Make it Visible: Using flip charts, handouts, 35mm slides, and other visual aids provides a welcome break in the presentation provided they are not overdone.

Use the Group: This method of communication uses the students to provide input for discussions.

Use an Expert: Although the expert you use may incorporate any of the above methods, this method may add more credibility to the presentation.

Test for Understanding

Once the information has been presented, the students must be tested to see if they understood what was communicated. Testing for understanding allows us to determine if our presentation level is too basic or advanced. It also gives us an indication to speed up, slow down, move on, or review what was just presented.

Practice

For the students to be successful, they must practice their newly learned skills. Practice comes in many forms but hands on exercises, where the student can actually use the learned skill, is the best.

Feedback

Students can practice all they want, but if they are practicing the skill incorrectly, they will not be successful. Therefore, the instructor must provide feedback to the student that tells them they are doing things wrong or confirmation that they are doing things right.

Although the above steps are listed in an order, it is important to understand that these steps are to be done on an ongoing basis. Motivation is not done once and then forgotten, we must continue to motivate the student during the entire learning process. More than one method of communication can be used at the same time, and consider using all methods during the life of a class. Testing for understanding should be done frequently to insure that learning is progressing.

QUESTIONS AND ANSWERS

Before asking questions of your students, it may be a good idea to plan out what questions to ask. It sometimes is easier to formulate questions if we think about what type of answer we want to solicit. The type of answers may come from the following list:

Nod of head, Raise hand
Facts, Figures, Names
Yes/No verification
Yes/No opinion
Short answer information
Short answer opinion
Detailed explanation
Argument with support and conclusions
Summary or evaluation
Value Judgment

Once we decide on the type of answer we want, we can formulate the type of question to ask. A question can be open, closed, or rhetorical. An open question is a question that allows the student freedom to answer in any way he chooses, and for any length of time. A closed question asks the student for specific information with no freedom to elaborate. Rhetorical questions are questions that are not meant to be answered. They can be used to introduce a new topic or bridge between two topics. One of the pitfalls of a rhetorical question is that someone may try to answer it.

A question can also be theoretical, application, or opinion in nature. A theoretical question may ask the student to answer based on certain bits of information, to draw from their prior learning and formulate a theory on how things should work. A question that is application in form, may ask the student to tell how things actually do work. An opinion question simply solicits a students opinion on a topic.

Based on this, we can ask an open application question, a closed theoretical question, an open opinion question, a rhetorical theoretical question, etc.

Thus, to formulate a question, consider the answer. If you want a Yes/No verification answer, ask a closed application.

If you are looking for a detailed explanation, ask an open theoretical question. For a student to give a value judgment, he would need to be ask an open opinion question.

QUESTIONING STYLES

You may want to ask several questions, one right after the other. If you do this, here are a couple of techniques you may want to try.

Extending: Asking the same type of question and looking for the same type of answer. This may be good if you want to do a brief review of earlier material.

Step-by-Step Up: Ask initially easy question, gradually getting harder with each one. This may be good if you want to find out what knowledge level a student is at.

Step-by-Step Down: Sameas above, but start with hard question and graduate to easy question.

Funneling: Start with very general open questions then gradually change to more specific closed questions. This style may be useful when introducing a new topic.

Sowing and Reaping: Start by asking specific closed questions, progress to more open questions, then go back to closed questions.

PUTTING IT TOGETHER

Now let's take a look at how we can use question in each of the five steps of our learning process.

Motivation

Rhetorical type questions that make the student think about deficiencies and how to overcome them may be useful for motivation. "Have you ever wanted to do such and such, but were not able to because of so and so?" for example. Try the "Extending" style and ask the same type of rhetorical questions to gain interest in the subject.

Another method of generating motivation that has worked for me is to ask an open application question. For example you may ask, "If you knew..., how would you use it?" This type of question will help show the student the "pay-off" of their learning, and because it is their idea, the pay-off is of interest to them. Another reason this is a good type of question to use is because it allows you to restructure your presentation to incorporate examples and comments that are of interest to the students. Doing so may help to maintain motivation and keep the pay-off in the students' minds.

Sometimes just asking "Why are you here?" or "What would you like to get out of this?" may help in motivation. I have often done this and wrote the students responses on a flip chart and then posted the flip chart pages on the walls of the room. It allows me to, at any point, point to a response and use it as a motivational tool.

Communication

Many types of questions can be used to communicate information to students. Rhetorical theoretical questions can be used to introduce material or as a transition between topics. Closed application questions that ask for short answer or facts can be used to summarize a lesson.

I like the "Use the Group" method of communication. With this method, open questions of any type can be used to direct the discussion or keep it moving. You may want to try the "Funneling" style of questioning when using the group. Start with open questions that ask for general answers. Then, as the group comes up with several ideas, choose the idea you want to elaborate on and ask questions that require a more focused or detailed answer. Then move to the more structured "State It" method of communication and complete the transfer of information.

Test for Understanding

This is where it gets very easy to ask "Any Questions?" If there is no reply, we conclude that the students understood what we told them. Try to stay away from this, rather ask the students some questions to see if they understood. Almost any questions you ask will test for some level of understanding. The "Step-by-Step Up" style of questioning is very useful in determining the level of understanding the students have.

Practice

Most practice of a new skill comes in the form of hands-on exercises that the students can perform. However, if you ask your students a few questions just prior to the exercise, they may be more successful in their practice. For example, if the students are to practice a new skill or set of skills, a few closed application questions that ask for facts or explanations may help them think through the practice session before starting. You may start with "In this practice session you will do..., what is the first thing you should try?", "What next?", "What is the important information you must remember?", and use an Extending or Funneling questioning style.

Feedback

You usually do not think of providing feedback by asking questions, but you can effectively use questions to set up feedback, and reinforce a student's progress. For example, in a technical class, you normally get a group of people with a wide range of knowledge. If you want to give positive feedback to one of the slower students, you may ask him a simple closed question you know he can answer, and then praise him for the correct answer. More difficult questions that require more detailed answers can be asked of students who are faster and require more of a challenge. In general, ask questions that are at the level of the students so you can reward them with positive feedback and encourage them to strive for more.

CONCLUSION

The goal of this paper was to discuss how we as instructors can enhance the learning process by the types of questions we ask. Firstly, we discussed a commonly used five-step method of teaching: Motivation, Communication, Test for Understanding, Practice, and Feedback. Next, we saw examples of different types of answers a student can give and different types of questions we can ask. Remember, to formulate a question, consider what type of response you want from the student. Lastly, we saw examples of how asking questions fit into each the five steps of our teaching method. Do not limit yourself to using questions in only one of the steps.

Students will benefit more from a training situation if they feel they have an active part in the learning process. One way we can get students involved and make them feel they have contributed is to ask questions in all areas of the learning process. Try it. It has worked for me and I believe it will work for you.

Graphics and the Information Center Manager

Effectively Integrating Business Graphics with Existing Information Processing Systems

Product Marketing
Personal Software Division
Hewlett-Packard Company

This document is intended to be used as a handout accompanying a 35mm or overhead transparency presentation. All of the graphics were created using the Hewlett-Packard Graphics Gallery Software. The pictures and clip art illustrations were obtained from several HP Graphics Portfolios.

Overhead transparencies were produced on the HP 7550 plotter. 35mm slides were produced on the HP 7510 film recorder. This handout was created in PageMaker by incorporating pictures from the Graphics Gallery and text that was created in HP Executive MemoMaker. The entire document with text and graphics was printed on a Hewlett-Packard LaserJet Plus.

Special thanks to Marilyn Ruel, Carol Luebke, Cathy Hughes and D.J. Jennings for their assistance in preparing this report.

This presentation is designed to provide Information Center Managers with an analysis of end-user needs for graphics as these applications have migrated from mainframes and minicomputers to PCs. It will explore some of the major issues faced by Information Center Managers as they integrate graphics into their existing information processing systems.

The presentation will also provide a checklist of items to consider in evaluating both graphics hardware and software. It will conclude with a brief discussion of future trends in graphics technology.

The power of graphics in improving communication*

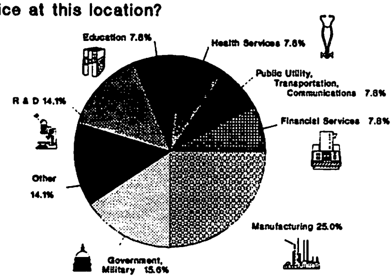
- Meetings tend to be shorter - by as much as 25%
- Presenters using overhead transparencies are perceived as:
  - More persuasive
  - More professional
  - More credible
  - More interesting

* As reported in the Wharton Applied Research Center Study.

HEWLETT PACKARD

Studies by the Wharton Applied Research Center and the University of Minnesota have confirmed that graphics can be a very effective method to communicate information.

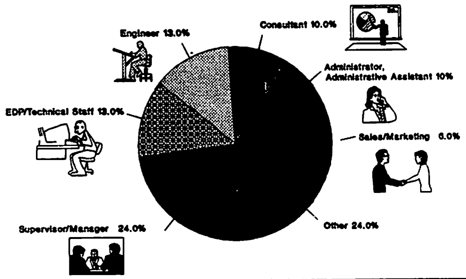
What category best describes the business or service at this location?



HEWLETT PACKARD

Hewlett-Packard conducts many on-going research studies to better understand user needs for graphics products. A diverse sample of companies was used in one of the most recent surveys.

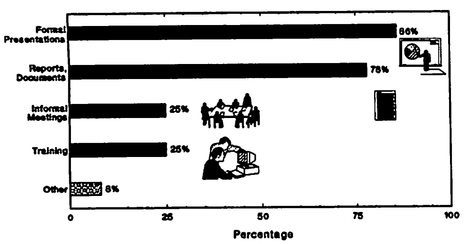
Which category best describes your job function?



HEWLETT PACKARD

Many different types of end-users were surveyed using a five page questionnaire.

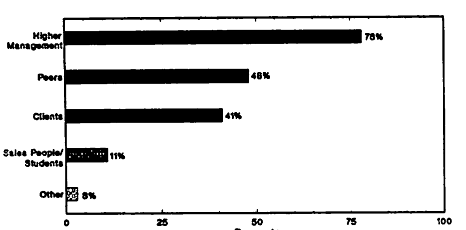
For what purposes will you use the graphics you create?



HEWLETT PACKARD

Formal presentations and documents were cited as the most frequent use for graphics.

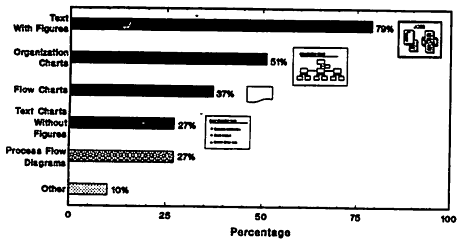
Who is the target audience for the graphics you create?



HEWLETT PACKARD

Not surprisingly, these graphics presentations were most often targeted at "Higher Management."

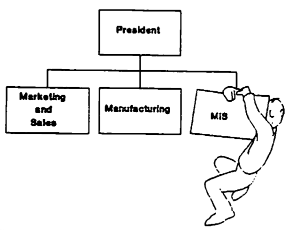
What kinds of graphics do you intend to create?
(besides the standard pie, bar and line charts)



HEWLETT PACKARD

Aside from the standard bar, pie and line charts, "Text Charts with Figures" was the most often mentioned type of chart. This is consistent with other studies which have shown that 50-75% of all graphs are text charts. Pictures are very useful in helping people to retain key points.

Do you sometimes feel like you are barely hanging on to your organization?



HEWLETT PACKARD

In fact, if you retain nothing else from this presentation, you will probably remember this slide because of the illustration.

What do users dislike in their graphics software?*

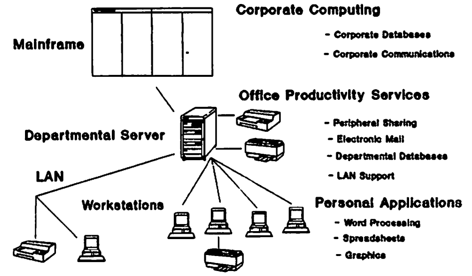
- 54% of people surveyed regretted their purchase because the software was:
  - too slow
  - too complicated
  - too hard to learn
  - "clumsy"
  - required excessive training
  - had bugs

* As reported in PC Week, December 23, 1986

HEWLETT PACKARD

After reviewing why people like to use graphics, it is also interesting to see what they dislike. A survey by PC Week uncovered an astonishing <u>54% of people who regretted purchasing their graphics software!</u>

Todays Computing Environment

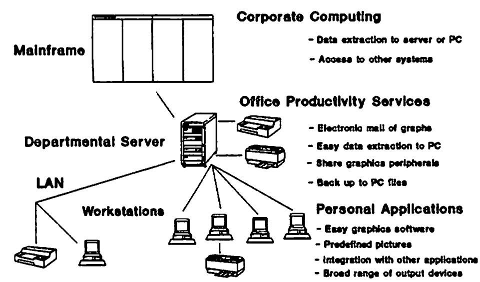


HEWLETT PACKARD

Undoubtedly, you have some combination of mainframes, minicomputers, LANs and PCs. The distribution of processing power down to the workstation level, and the quality and quantity of applications on PCs, have been two of the driving forces in the development of a three tier, information processing environment.

Today, it is much more cost effective to put CPU intensive applications, such as graphics, on workstations where the cost per MIP is about $ 10K vs. $ 70-75K on a minicomputer and $ 150K on a mainframe.

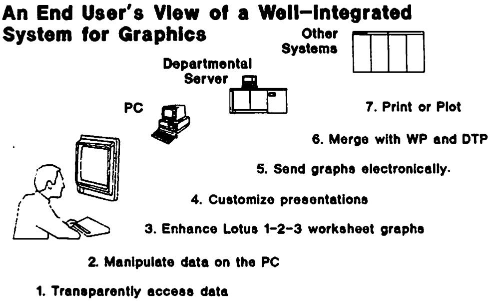
Graphics Users' Needs In Todays Computing Environment



HEWLETT PACKARD

A person developing graphics on a Personal Computer should not be limited to functioning as one individual working in a vacuum. Instead, he should be able to take advantage of the benefits that are provided by a well integrated three tier system.

An End User's View of a Well-Integrated System for Graphics



HEWLETT PACKARD

In fact, from a users perspective, the PC should act as a window to the resources that are spread throughout the three tiers. In this way, a user should be able to retrieve data from remote systems, manipulate it in his PC applications, turn it into graphs, send the graphs to others, merge them with text in documents and print or plot them on a variety of output devices.

Five Reasons to Standardize on Graphics Software

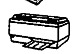
- ☐ Guarantee graphs and pictures can be shared
- ☐ Reduce the 54% of users who regret their purchase
- ☐ Ensure integration with existing information systems
- ☐ Eliminate the need for retraining later
- ☐ Reduce costs through site licenses

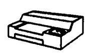
HEWLETT PACKARD

To really exploit the capabilites of your information processing systems, and to maximize the benefits to your users, you may wish to standardize on one graphics software package.

PC Graphics Hardware: Checklist

- ☐ Intel 80286-Class Machine
  - Color Monitor
  - Hard Disc
  - Printer
  - Plotter





Optional: 35mm film recorder, video output devices, color printers.

HEWLETT PACKARD

Depending on your users specific needs, you will probably want to consider an 80286 class machine with a hard disc and color monitor as a graphics workstation. In the survey conducted by Hewlett-Packard, over 80% of respondents had such a system.

PC Graphics Software: Checklist

- ☐ Graphical interface for ease-of-use.
- ☐ High quality output.
- ☐ Choice of professional quality fonts.
- ☐ Ability to enhance Lotus worksheet graphs.
- ☐ Standard charts: pie, bar, line, scattergrams.
- ☐ Ability to enhance standard charts with pictures.

HEWLETT PACKARD

In selecting software, you should try to choose a package that is easy to use, as well as being highly interactive, in order to avoid the possibilty of end-user rejection.

PC Graphics Software: Checklist

- ❑ Extensive portfolio of pictures and symbols.
- ❑ Intelligent support of peripherals.
- ❑ Integration with desktop publishing and image scanners.
- ❑ Training: classroom, computer assisted, self-paced ...
- ❑ Vendor support: phone-in, on-site ...

HEWLETT PACKARD

In addition to the standard features, you should also consider how well the software exploits the hardware and peripherals' capabilities. Vendor support and training programs that meet the needs of your organization are also important variables that should be considerd to ensure a smooth, effective implementation of graphics systems.

PC Graphics Software: Desirable Features

- ❑ Additional chart types: linear regression, combination bar/line, surface, 3D, double Y axis ...
- ❑ Painting and free-hand drawing.
- ❑ Interchange of graphs with minicomputers and mainframes.
- ❑ Ability to add customized logos and symbols.
- ❑ Availability of software in site licenses.

HEWLETT PACKARD

In addition to the standard features, there are a number of desirable features you may wish to consider in your selection of software.

Future Trends

- ❑ Lower price/higher quality laser printers
- ❑ Better quality color printers
- ❑ Higher resolution graphics displays
- ❑ Broader selection of high quality fonts
- ❑ Graphical interfaces with pull-down menus
- ❑ Better integration of graphics with Desktop Publishing
- ❑ Emphasis on graphics integration in the workgroup

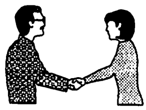
HEWLETT PACKARD

In the future, you can expect to see higher quality and lower cost color output devices as well as faster and higher resolution monitors. These high resolution monitors, the use of a mouse, and graphical user interfaces with pull-down menus are quickly becoming standards in graphics systems.

Essentially, graphics will become just another integrated tool that can be employed by an entire workgroup to more effectively communicate information and decisions.

Integrating Business Graphics with Your Information Systems Can:

- ☐ Maximize end-user satisfaction
- ☐ Reduce end-user frustration
- ☐ Increase utilization of existing investments
- ☐ Optimize system resources



HEWLETT PACKARD

Well-integrated business graphics can increase user satisfaction with information processing systems and optimize the utilization of existing investments.

If you have any questions about the contents of this presentation, or if you are interested in learning more about graphics products and how they can be integrated into your information processing environment, please contact me or your local Hewlett-Packard representative or authorized retailer.

Thank you,

Chris Kocher
Product Marketing Manager
Hewlett-Packard
3410 Central Expressway
Santa Clara, CA 95051

MPE XL Mapped Files

Bryan Carroll
Computer Systems Division
Hewlett Packard
Building 44-MV
19111 Pruneridge Avenue
Cupertino, Ca. 95014

Introduction

The MPE XL operating system introduces many new, exciting and powerful features to make programs more efficient and perform more effectively. Perhaps the single most powerful and exciting new feature of MPE XL is User Mapped File Access. User Mapped File Access can increase the efficiency of an application causing it to run many times faster than it would without User Mapped File Access. This paper will explore how mapped files are implemented, the performance gains possible, and some uses of User Mapped Files to replace MPE V features and increase application performance.

User Mapped Files are possible because of the HP Precision Architecture's expanded 64 bit address space. A Mapped File is a disc file that is mapped directly into the virtual address space. User Mapped Files are managed with pointers that are returned from a new system intrinsic, HPFOPEN, which has similar functionality to the existing MPE V FOPEN. The use of pointers allows a programmer to view a Mapped File as if it were a large array. Once the Mapped File is open, the file system is not needed to access the file until you are ready to close the file. As a program moves through a file it will eventually access a part of the file which is not in memory. The memory manager will be invoked to bring the missing part of the file into memory so it can be accessed by the program.

Impressive performance gains have been observed with Mapped Files. With the use of pointers, all file system overhead can be bypassed which increases performance. User Mapped File Access essentially replaces MPE Disc Caching functionality in many cases without the overhead of the file system or Disc Caching software which further increases performance.

Mapped Files

User Mapped File Access is available directly to MPE XL users without special capabilities via the HPFOPEN intrinsic (new with MPE XL). User Mapped File Access is also available to the various modules in the operating system and, in a way, all files on the system are accessed as Mapped Files. The difference between User Mapped Files and Mapped Files accessed through the file system is who controls the pointer to the file. If a file is accessed with the file system intrinsics (FREAD, FWRITE, etc), the file system will maintain the pointer to the file. If a file is opened with HPFOPEN specifying the appropriate options, a pointer is returned to the caller who can access the file by referencing the pointer. A file can be opened with User Mapped Access and accessed with a pointer as well as accessed with the file system intrinsics. In this case, there are two separate pointers for the file, one maintained by the file system and the other maintained by the user program.

MPE XL File System

The major advantage in using User Mapped File Access is the performance benefit that can be realized in bypassing the file system. Let's take a look at some of the tasks performed by the MPE XL File System so we can better understand the size of the performance gains possible with User Mapped File Access.

File System Example - FREAD

Let's examine a commonly used intrinsic like FREAD. The path through FREAD can be broken down into three major sections; the FREAD Intrinsic itself, the Type Manager, and the Storage Manager.

The **FREAD** intrinsic is responsible for the usual checking of parameters that takes place for all intrinsics. FREAD must also obtain the Process Local File Descriptor (PLFD) entry for the given file. The PLFD is like the MPE V Active File Table (AFT) which keeps track of all open files. The FREAD intrinsic must also verify your access rights to the file each time the intrinsic is called.

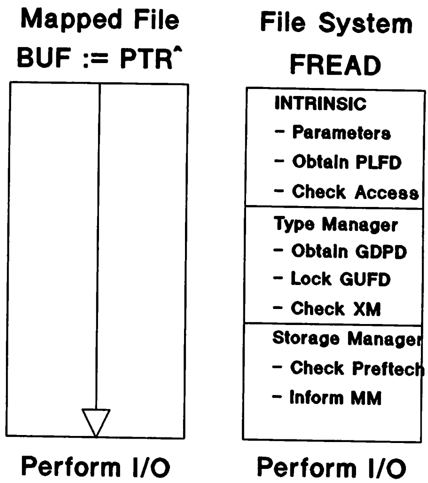
The **Type Manager** is a new concept within MPE XL. There is a Type Manager for each specific type of file such as Fixed Record Length disc files, or Variable Record Length disc files. The Type Manager is responsible for obtaining the Global Data Pointer Descriptor (GDPD), which is very similar to the virtual address used for User Mapped Access which we will discuss later. The Type Manager is also responsible for locking the Global Unique File Descriptor (GUFD), which is like a combination of File Control Blocks (FCB) used on MPE V. The final task of the Type Manager is to check the connection between this file and another new MPE XL concept, the transaction manager. The transaction manager is an operating system subsystem which provides data integrity to many disc resident structures including selected user files.

The third section of the FREAD path involves another new concept, the **Storage Manager.** There is a unique Storage Manager for each type of physical storage device such as discs, tapes and printers. In the FREAD path, the storage manager is responsible for determining if a prefetch must be performed. The Storage Manager is also responsible for informing the memory manager when an I/O must be performed. The memory manager is responsible for all I/O on MPE XL systems.

The prefetch defined by the Storage Manager comes in two varieties: hard prefetches in which a process must block, and soft prefetches which do not require a process to block. The Storage Manager will prefetch from two pages (8K bytes) to a maximum of 64 pages (256K bytes) depending on the method of access (random or sequential). The Storage Manager optimizes the prefetch very well. If four consecutive reads take place to consecutive addresses in the file, the Storage Manager will recognize this as sequential I/O and perform a prefetch of 32 pages EVEN IF YOU ARE USING FREADDIR!

User Mapped Files

User Mapped File Access bypasses all of the above file system code and allows the user to directly access file pages. A file page is just a main memory copy of a 4K byte portion of a Mapped File. User Mapped File Access works by referencing the file as an array. A reference to a part of the file that is not already in memory will result in a page fault and the memory manager will bring the needed file page into memory, just as when a fault on a code page causes the memory manager to bring in the needed code page.



Mapped Files vs. File System

The performance gains possible with User Mapped Files when compared with using the MPE XL File System vary depending on the operation. In general, the CPU time required to perform a set of file operations will always be less when using User Mapped Files. The elapsed times however, will vary depending on the access (random or sequential). In general, the elapsed times for random access using User Mapped Access will be less than when using the file system. The elapsed times for sequential access of large files is often longer when using User Mapped File Access because of the prefetching the MPE XL File System can perform. Consider the following two examples:

Example 1 - Random Reads

Environment: Native Mode Pascal/XL program, 2 HP7933 Discs, 25000 record (100 MB) file, blocking factor = 1.

Test: 500 Random Reads

Results:

	CPU	Elapsed
User Mapped File:	913 ms	9958 ms
MPE XL File System:	1629 ms	28459 ms
Performance Gain:	78%	186%

Example 2 - Sequential Reads

Environment: Native Mode Pascal/XL program, 2 HP7933 Discs, 25000 record (100 MB) file, blocking factor = 1.

Test: 25000 Sequential Reads

Results:

	CPU	Elapsed
User Mapped File:	41017 ms	917393 ms
MPE XL File System:	73277 ms	307447 ms
Performance Gain:	79%	-66%

MPE Disc Caching

User Mapped File Access is a new and improved implementation of MPE V Disc Caching for some situations. The overhead involved with searching cache domains and mapped entries is eliminated when compared with User Mapped File Access.

The concepts of User Mapped File Access are very similar to MPE V Disc Caching except for the reduced CPU overhead of Mapped Files, and the prefetch or domain size is fixed at 4K bytes. When performing random access, the reduced CPU overhead allows User Mapped Files to perform better than all other access methods as illustrated in Example 1 above. Despite the larger prefetches available through the file system, User Mapped File Access can sometimes outperform the file system even when performing sequential operations because of the greatly reduced CPU requirement of User Mapped File Access.

Languages

User Mapped File Access requires the use of a "pointer type" variable and therefore can only be used with languages that will support a pointer type variable. User Mapped File Access is also not available in compatibility mode. The only languages that currently meet these requirements are HP Pascal/XL and HP C/XL. A possible third alternative is the SPL language when used with the native mode SPLash compiler since SPL supports a pointer variable type. Please contact the SPLash vendor, Software Research Northwest, for more information about the use of SPLash with User Mapped File Access.

It is possible to use User Mapped File Access from languages that do not support a pointer type variable by using procedures, functions or subroutines. A routine could be written in a language that does support the pointer type variable like Pascal/XL which could then be called from the main application which could be written in a language without pointer types like COBOL/XL or FORTRAN/XL. This option should be carefully considered since it adds complexity to the design of the application and therefore reduces supportability. This added level of complexity may not be worth the potential increased performance of User Mapped File Access.

HPFOPEN

The new MPE XL File System Intrinsic, HPFOPEN, provides access to the file pointers needed by programs wanting to use User Mapped File Access. Although the FOPEN intrinsic is still available, the new HPFOPEN intrinsic is a native mode superset of FOPEN and should be used in all native mode applications.

The new HPFOPEN intrinsic uses the following format.

```
HPFOPEN(FILENUM, STATUS, itemnum, item,
                         itemnum, item,
                            .       .
                            .       .
                            .       .
                         itemnum, item);
```

The itemnum/item pairs replace the positional FOPEN parameters and are used for specifying all file open criteria to HPFOPEN. A complete list of the item numbers, which currently range from 0 to 59, is available in the MPE XL Intrinsics Reference Manual (Part Number 32650-90028). There are item numbers for all FOPEN parameters like record size, file code and blocking factor, as well as many new parameters like the 'will access' parameter to specify your anticipated access (Random or Sequential), and the 'long mapped' option for use with User Mapped Files.

User Mapped Access HPFOPEN Parameters

There are only two HPFOPEN parameters that are specific to User Mapped File Access, Long Mapped and Short Mapped. Since all files are really mapped files, the only thing we must do to receive mapped access to the file is to obtain the pointer to the file. This ability is provided in two different forms; a 64 bit long pointer (Long Mapped) and a 32 bit short pointer (Short Mapped). Specifying either of these two parameters to the HPFOPEN intrinsic will return the appropriate length pointer to your program. Opening a file with either of these mapped parameters does not prevent you from using any of the file system intrinsics like FREAD and FWRITE. File system intrinsics and User Mapped File Access can be used together by the same application.

Once the file is opened by HPFOPEN using either the Long or Short Mapped options, data can be read from or written to the file by indexing off the pointer. The pointer returned by HPFOPEN points to the first byte in the first record of the file. Unpacking of records in a variable length or undefined length record file must be performed by the user program. If the file has user labels, these can be accessed by negatively indexing from the pointer returned by HPFOPEN.

The following example contains portions of a Pascal/XL program which randomly reads and writes to a file using User Mapped File Access:

```
const page_len          = 4096;
type  page_type         = packed array [1..page_len] of char;
      file_pointer_type = ^ $extnaddr$ page_type;
var   base_file_ptr   : file_pointer_type;
      file_ptr        : file_pointer_type;
      rec_num         : integer;
      num_recs        : integer;
      filename        : packed array [1..38] of char;
      perm_file       : integer;
      update_access   : integer;
      buf             : page_type;
```

```
begin
        .
        .
        .

HPFOPEN(file_num, file_status,
        2, filename,
        3, perm_file,
       11, update_access,
       21, base_file_ptr);  { Long Mapped Pointer }

        .
        .
        .

file_ptr := addtopointer(base_file_ptr, rec_num * page_len);
if read_only then
   buf := file_ptr^;  { Read record 'rec_num' into buf }
   else
   file_ptr^ := buf;  { Write buf into record 'rec_num' }

        .
        .
        .
```

In the above example, a file is opened with the HPFOPEN intrinsic and depending on the 'read_only' flag, data is written to or read from the file using the Long Mapped pointer. Part of the power of User Mapped File Access can be seen in this example. The reading and writing of the file is performed by the memory manager when the pointer is simply referenced in an assignment statement. Notice in the example that the pointer returned by HPFOPEN which points to the first byte in the file is retained and a second pointer is used to move through the file.

Programming Considerations and Limitations

Most new features and enhancements have tradeoffs and it is no different with User Mapped File Access. The benefits of bypassing the file system with User Mapped Files also has the cost that some file system operations must still be performed and are left to the user. Designers and Programmers must be aware of several considerations and limitations involving the use of User Mapped Files. We have already discussed the Language considerations and will now introduce other considerations and limitations one at a time.

End of File Pointer

One of the benefits of using User Mapped Files is that the file system overhead can be eliminated. Some functions that the file system would normally perform, like maintaining the end of file (EOF) pointer, must still be performed. Since the file system is not normally called when using User Mapped Files, the user must maintain the end of file pointer. Anytime data is added to a file beyond the current end of file pointer, this data will be lost unless the program also moves the end of file pointer. The end of file pointer can be maintained with the FPOINT and FCONTROL intrinsics as in the following example.

```
const page_len          = 4096;
type  page_type         = packed array [1..page_len] of char;
      file_pointer_type = ^ $extnaddr$ page_type;
var   base_file_ptr  : file_pointer_type;
      file_ptr       : file_pointer_type;
      new_eof        : integer;
      current_eof    : integer;
      num_recs       : integer;
      filename       : packed array [1..38] of char;
      perm_file      : integer;
      update_access  : integer;

begin
        .
        .
        .

hpfopen(file_num,file_status,
         2, filename,
         3, perm_file,
        11, update_access,
        21, base_file_ptr);
        .
        .
        .

{ Get the current EOF }
flabelinfo(filename, 0, error, itemnums, items, itemerrors);
        .
```

```
        .
        .

new_eof := current_eof + num_recs;  { Compute new EOF }
fpoint(file_num, new_eof);          { Move the File Pointer }
if ccode <> cce then error;

fcontrol(file_num, 6, dummy);  { Post the new EOF }
if ccode <> cce then error;
```

In the above example, a User Mapped File is opened and 'num_recs' records were added to the end of the file. In order to preserve these records, the end of file pointer had to be moved out 'num_recs' records which is performed by moving the current record pointer with FPOINT and posting the end of file pointer at that point with the FCONTROL intrinsic. If the end of file pointer had not been moved by the program which added records to the User Mapped File, the new records would have been lost since they were added to the file beyond the file limit.

User Mapped Files and File Types

Opening a file with User Mapped File Access allows the user to manipulate the data within the file, including any file structure information like end-of-record markers in variable length record files without restriction. In order to preserve the integrity of file types that incorporate file structure information in the file along with the data, User Mapped File Access will not be granted to certain file types. File types restricted from User Mapped File Access include Relative I/O files (RIO), Circular files (CIR), Message files (MSG) and any non-disc files. Files with variable length records and KSAM files can be opened as User Mapped Files only if they are also opened with read only access.

Concurrent User Mapped File Access

When a User Mapped File is opened with a Short Pointer, space for that file is allocated from a finite set of virtual addresses that I will refer to here as **Short Pointer Space.** This space for Short Pointers with finite capacity is a central pool of space for the entire system. Because it is a finite resource, its usage is limited to a maximum of four megabytes per file opened with a Short Pointer, and a maximum of six megabytes of Short Pointer Space used for User Mapped

Files at any one time. A four megabyte file is equivalent to a file holding about 15000 sectors while a limit of six megabytes for all files opened with Short Pointers is equivalent to files consuming about 23000 sectors.

An additional consideration regarding pointers is also related to the finite capacity of Short Pointer Space. If a file is first opened as a User Mapped File with a Long Pointer, any subsequent attempt to open the file as a User Mapped File with a Short Pointer will be denied because the file cannot be re-mapped into Short Pointer Space. If, however, the file was first opened as a User Mapped File with a Short Pointer, it could be opened as a User Mapped File with a Long Pointer with the above file size constraints. The default for both HPFOPEN and FOPEN is to open a file with a Long Pointer. Once a file has been opened with HPFOPEN (using defaults), FOPEN or User Mapped File Access with a Long Pointer, it cannot be opened with User Mapped File Access using a Short Pointer until all accessors have closed the file.

Protection ID's

Another consideration involving User Mapped File Access involves the way the memory manager keeps track of the pages that are in main memory. The memory manager maintains a table called the Page Directory (PDIR). There is one entry in the Page Directory for each page in memory. Each entry contains information about the page such as its disc address and a Protection ID or PID. Every time a page is referenced (such as when a record in a User Mapped File is read or written), the page directory is searched for the referenced page. This search is performed in hardware and is very fast (usually completing in 1/2 of one machine cycle). Once the entry is located, the Protection ID is compared to a cache of Protection ID's for the process which has just referenced the page. If the Protection ID is not found, a software routine must be invoked (an expensive operation) to determine if access to this page can be granted to this process. Since the Protection ID's are associated with the process, multiple processes that share pages with other processes or access a large number of pages randomly will require the software routine to be invoked frequently to determine if access can be granted to the new process. This software routine will add the Protection ID to the processes Protection ID cache and remove the oldest entry in the cache. Consider the following example:

Assume a process was accessing a User Mapped File. Assume that this process also accessed many other memory structures such as code objects, data objects and other files. When the process began, it would access the Mapped File and the file's Protection ID would be added to the Protection ID cache for the process. If the process continued to access other memory structures, it is possible to create a situation where system software would have to be invoked every time a page was referenced to add the new Protection ID to the user Protection ID cache. This Protection ID thrashing could degrade system performance when compared to using the file system since the file system does not use the users Protection ID cache. The savings of bypassing the file system however, will usually allow improved performance when compared with access through the file system. We have not seen any cases where shared User Mapped Files have degraded performance, but the potential is there at least in theory.

Applications for User Mapped Files

There are many applications for User Mapped Files. This efficient access method can be used to enhance the performance of many applications. The most obvious application for User Mapped Files is when a new application is being designed that requires random access. User Mapped Files could be implemented to store and retrieve data quickly and efficiently.

User Mapped Files Replace Extra Data Segments

When migrating an application from MPE V to MPE XL, there are several MPE V features which have been duplicated in MPE XL, but may not be as efficient as another MPE XL feature. MPE V Extra Data Segments (XDS) is one such feature that has been provided in MPE XL, but it is still in compatibility mode and the original MPE V design remains which prevents it from performing as well as it could. One use of User Mapped Files is to replace the Extra Data Segment procedure calls with calls to user written native mode procedures that access User Mapped Files.

One approach to this problem would be to write several procedures with the same names as the Extra Data Segments (GETDSEG, DMOVIN, DMOVOUT, FREEDSEG, ALTDSEG). These procedure could then be added to a library that is referenced by the main program which uses the Extra Data Segment Intrinsics. The translation could be made between what the Extra Data Segments expect and a User Mapped File in the user written procedures. These procedures could be adapted to work with several different applications and even third party applications or applications where source code is not available.

Summary

User Mapped Files are a powerful tool to add to the programming and design tool kit. As with most new features, there are drawbacks, but in most cases, the drawbacks are far outweighed by the benefits of User Mapped Files. Impressive performance improvements have been observed from a variety of applications running on MPE XL. User Mapped Files is one benefit of the new Hewlett Packard Precision Architecture which will prove very beneficial.

Help! Stack Overflow!
Alternatives for reducing stack size on MPE V/E HP 3000s

Lisa Burns Hartman
Hewlett-Packard
Corporate Offices
P. O. Box 10301
Palo Alto, CA 94303

STACK OVERFLOW! Those two words cause even the most experienced HP3000 programmers to groan and shake their heads. Writing a large application and making it work within the HP3000's stack limit of 32K words can be a real challenge. In this article I will suggest some techniques for handling large amounts of data within the constraints of the 3000's stack architecture.

What can cause a Stack overflow condition?

When a program aborts with a STACK OVERFLOW message, it means that the 32KW limit for stack size has been exceeded. If you are running a brand new program and you encounter this message, it means that you have simply placed too many variables on your stack at once. For a COBOL program, this means that your Working Storage is too large for the HP3000 to handle.

But what if you are maintaining an existing application which has run successfully for some time and is now getting a STACK OVERFLOW message? There are several possible reasons this might suddenly occur. Opening additional databases or files will increase your stack size. Each additional open adds between four and 16 bytes to this area. Another programming change which may increase stack size would be a VPLUS form change adding additional fields. VPLUS itself will use additional buffer space for each added field, in addition to the necessary additional room within your program's screen buffer.

Perhaps the most likely reason for an increase in stack size, however, is a new call to an additional subroutine. The linkage area necessary to communicate with the subprogram will add permanent space to the calling program. The local variables for the subprogram will add to the stack size as well. If the subprogram is static, this additional stack space will stay on the stack for the life of the process. If the subprogram is dynamic, local variables will pop off the stack as soon as control is returned to the calling program.

But, you protest, I haven't touched the program! All I did was bring up a new MIT, and now the application blows up. If your application was running very close to the stack limit on one MIT, and you upgrade your machine to the new MIT, this too can produce a stack overflow. MPE intrinsic calls (file intrinsics, IMAGE calls, etc.) are frequently changed for a new MIT. Just as for your own programs, their data usage may go up as well. For example, an FOPEN on UMIT takes more space than it did on T-MIT. Since all application programs call system intrinsics, changing a MIT can produce stack overflows in programs which ran successfully on the previous MIT.

Stack overflow: Quick fixes

Now that you have discovered the stack overflow, you have several options, some easy to implement, some harder. The first thing to try is running your program with the MAXDATA parameter.

```
:RUN MYPROG;MAXDATA=20000
```

The effect of this parameter is to increase the default stack size for your program. The default stack size is set by the compiler and may be inadequate for your program. It will certainly be inadequate for your program if you are using VPLUS screens and intrinsics. Figure 1 shows a diagram of the HP3000 stack structure. VPLUS intrinsics place data in the DL to DB area of your stack. Compiler defaults for the DL to DB area do not allow for this needed additional space. The MAXDATA parameter is therefore required to expand this area for VPLUS applications. Note that the STACK parameter will not expand the DL to DB area and so will not alleviate stack problems for VPLUS users.

The maximum value for the MAXDATA= parameter is 32000. The real word count of this stack area is 31232, but MPE is nice enough to take 32000 and not make you remember that. Setting the value to the maximum will not cause harm, since the stack space will be allocated as needed in 1K increments. Thus, you may wish to go ahead and set MAXDATA to 32000. If you would like to be more conservative, however, and have an idea of how much stack size you really need for your application, experiment with values until you find a minimum value which still avoids the stack overflow. You can do this by beginning with MAXDATA=32000 and working down by 5K increments, MAXDATA=31500, and so on, until you encounter an overflow.

If you are fortunate enough to find a value of 32000 or less which removes the stack overflow, congratulations! I recommend that you re-PREP your program with this same MAXDATA option before releasing it to your users. This will set the stack size for your object code to the MAXDATA value permanently, so that if your user forgets to RUN your program with the MAXDATA option, you will still be safe. Update your PREP job to include this option, so that you will also be safe for future program updates.

But what if you already have MAXDATA set at 32K, and you are still getting an overflow? The second quick fix is to execute your program with the NOCB (no control block) option:

```
:RUN MYPROG;NOCB
```

The PCBX, or Process Control Block eXtention, is used by MPE to manage the files and file equations used by your program (see figure 2a). This area can be very large if your program opens multiple files. The effect of the NOCB option is to move the PCBX area of your stack out to an extra data segment, freeing up more space for your application program's data area (see figure 2b). Be aware, however, that this will mean another data segment which must be CPU resident in order to execute your program. If you are running on a machine which is memory-bound, running your program with the NOCB option may increase memory thrashing and degrade your application's performance. Also, another data segment means another DST (data segment table) entry. For MPE IV and earlier MIT's, the DST is limited to 192 entries. For machines running these MIT's, the NOCB option may cause the error "OUT OF DST ENTRIES". MPE V machines will not be affected by an additional DST entry, since the DST on these machines can be configured for up to 2048 entries. So, for most newer machines with adequate memory running MPE V, the NOCB option poses no threat and may be a quick solution to a sudden stack overflow.

Harder fixes

Suppose that you have tried the above two methods and are still encountering an overflow. You're going to have to work a little harder to solve your problem. The next thing to do is to look at your subprograms. Static subprograms, $CONTROL SUBPROGRAM in COBOL, place data on the stack for the life of your process (see figure 3). This can be a problem, especially if their data areas are large. Dynamic subprograms, $CONTROL DYNAMIC in COBOL, free up stack space once control is returned to the calling program (see figure 4).

There are some things to be careful of when using dynamic subprograms. Remember that since local data areas disappear once a subprogram is exited, care must be taken that any permanent data be passed back to the calling program. Also, initialization routines will need to be executed each time a call is made to the dynamic subprogram.

Subprograms can have another effect on stack space. What if your main program, PROG A, calls a subprogram, PROG B, which then makes its own call to a third program, PROG C? What effect will nesting these subprograms have on your process's stack? Figure 5 illustrates the effect of this nesting. At the point it time when PROG C is being executed, the data areas of all three subprograms will be resident on the stack. This is true whether or not PROG B and PROG C are dynamic. Deep nesting can thus greatly affect stack space. This is another area to examine when checking your subprograms.

Still no luck, huh? You changed your subprograms to dynamic, you eliminated excessive nesting of subprograms, and you are still aborting with a stack overflow. It's time to look at your data areas themselves. Do you duplicate data in several areas in your global data areas (working storage and linkage in COBOL)? Are you passing more data than is necessary to called subprograms? Structured analysis and design techniques can help you identify necessary data flows. Taking the time to examine what exactly is needed by a called module instead of simply passing that variable called "01 Kitchen-sink-data-area" can help reduce stack space.

Hardest fixes for big problems

Some applications are just plain big and need just plain big data areas. If you suspect that this is your situation, you are going to have to work still harder. The first thing to consider is storing needed data outside of your stack. This can be done using extra data segments or using MPE temporary files. In either case, the program which needs access to the externally stored data will have to work harder than if the data were available directly.

An extra data segment (XDS) is an unstructured block of memory associated with your process. It can be used for large data areas, like a report page which is being formatted all at once, or for a table structure which occurs repeatedly. Programs accessing XDS must have the special capability PH enabled. Programming with XDS requires XDS intrinsics, GETDSEG, FREEDSEG, DMOVIN and DMOVOUT. The data segment is created with GETDSEG, and then loaded with data via the DMOVOUT intrinsic. Once loaded, the data in the XDS cannot be accessed directly. To access it, the programmer must bring the data into the stack using the DMOVIN intrinsic, and then manipulate it within the stack (see Figure 6). Since the DMOVIN and DMOVOUT intrinsics work with byte addresses, the programmer must keep an accurate count of where data is located within the XDS. Finally, the programmer should destroy the XDS with the FREEDSEG command. This will avoid problems with creating a data segment which already exists if the program is run a second time from the same session.

The advantages of XDS use is its speed. Since externally stored data is memory-resident, access is very quick. There are some disadvantages, however, and one is the programming complexity mentioned above. An additional wrinkle is that if your data area will not fit in one XDS, which has a maximum size of 32KW, you may need to work with several. This will further complicate your programming. Also, since the use of an XDS means an additional data segment for every user of the application program, this technique has the same problems as the NOCB option. But for MPE V machines with adequate memory, the use of XDS is a good solution.

An alternative to XDS use is the use of MPE temporary files for external data storage. Temporary files have advantage that programming is very simple. Programmers are typically familiar with file intrinsics and are comfortable using them. Like XDS, temporary files can be used for large data areas or for tables. Using the command intrinsic and the BUILD command, the temporary file can be built within the program. Simple reads and writes are used to access the data. The temporary file will be destroyed when the process quits, or it can be destroyed with the PURGE command within the program before the process terminates.

Temporary files have the additional advantage that they are not limited to 32KW and can be expanded as needed. And with disc caching enabled, performance is comparable to XDS use. This is because the cache domain for the temporary file will be memory resident, and reads and writes will be done through memory transfer and will be very fast. However, if you cannot guarantee that your program will run on a machine with caching enabled, MPE file access will significantly affect your program's performance, since accessing the data stored in the file will mean waiting for disc I/O. In this case, you should stick with XDS.

A final technique for solving stack overflow problems is Process Handling. Process Handling (PH) capability allows a process to RUN another program by creating a child process. The child process has its own stack and is independent of its parent (see Figure 7). PH can be used to treat a standalone subroutine, a print program for example, as a separate process. Like XDS, PH has its own set of intrinsics. The parent (calling) program uses the CREATE intrinsic to set up the child process, and then initiates its execution with the ACTIVATE intrinsic. At this point the parent program may use the SUSPEND intrinsic to stop its own processing until the child's function is complete. When the child has completed its task, it wakes up the parent with the ACTIVATE intrinsic, and then SUSPENDs itself. Finally, when the child will not be called again, the parent process destroys the child using the TERMINATE intrinsic.

PH has the advantage that the new child process gets its own stack -- another 32KW of space. There are some significant disadvantages to PH, however, the worst of which is probably the programming complexity described above. Care must be taken to synchronize the parent and child as they ACTIVATE and SUSPEND each other. If this logic is incorrect, data may be lost, or worse, both processes may be SUSPENDed at once, so that nothing will happen at all! It should also be noted that since there is noticeable performance overhead on the first call to the child, the child process should not be TERMINATEd until it is clear that no further calls are needed.

Programming with PH is further complicated by the fact that since the child process is independent of its parent, it must perform its own database opens and file opens, even if the parent has already opened these files. It will then have its own file pointers and database control blocks. And since it cannot make use of linkage areas or passed variables like a subprogram can, the child must use inter-process communication techniques such as XDS or Job Control Words (JCW's) in order to pass data back to the parent.

A final caution about PH is that it will double the number of processes running for a given application. This must be taken into account as far as DST use and PCB entries are concerned, especially on MPE IV and earlier MIT's. Database applications must also remember that there will now be twice as many processes accessing a given database, which may affect IMAGE logging and database locking strategies.

How to avoid stack overflows in the future

The best way to avoid being surprised by stack overflow is to know how much stack you are using in all areas of your application. The process display within the process context of OPT.PUB.SYS can show you the approximate stack usage for a given program. By running the program on one terminal and monitoring the stack use on another, you can see stack size change as you perform different functions within your program.

Figure 8 shows an example of the process display. The process shown has at one point used 27648 words of stack (SIZE). This is its "high-water" mark, the largest the stack has been. Its PCBX area is 1329 words (SYSOV). Its VPLUS area is 7640 words (DL-DB). Its main program has a data area of 10967 (DB-QI), and it has one subprogram with a data area of 3924 words (QI-Q). To determine how large its stack is at present, we need to subtract 3779 (S-Z), the space between the current stack pointer and the high-water mark, from the high-water mark, 27648 (SIZE). Thus, the current stack usage for this program is 23869.

An alternative to OPT is available for programs which do not call VPLUS intrinsics. For these programs, adding the data areas your main program and subprograms will give you a close estimate of stack space. This data area is shown in octal at the end of each compile listing.

Keep in mind, however, that this total will not reflect the PCBX portion of your stack.

Once you have established your current stack size, be aware of how programming changes and enhancements will affect stack use. Increases in linkage areas and global areas will add space, as will new called modules and additional file opens. These increases must be considered as you design these changes.

If you are dangerously close to the magic maximum value of 32636 words (the PCBX area plus the MAXDATA area) take steps NOW to overcome the problem, not when you blow up! Consider using XDS or temporary files. Rewrite called subprograms as process handled programs. Break up one program into several if possible. Buy yourself some room for growth. You will sleep better at night if you know that your maximum stack use is 22000, not 31999 words.

Of course, the real answer to stack space is to port to Spectrum, where the 32K word limit will be a thing of the past. I can't wait!

HP3000 Data Stack

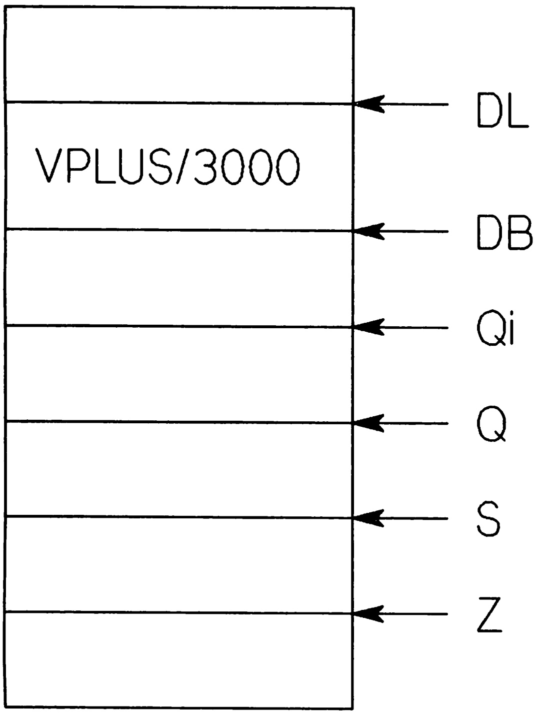


Figure 1

:RUN MYPROG

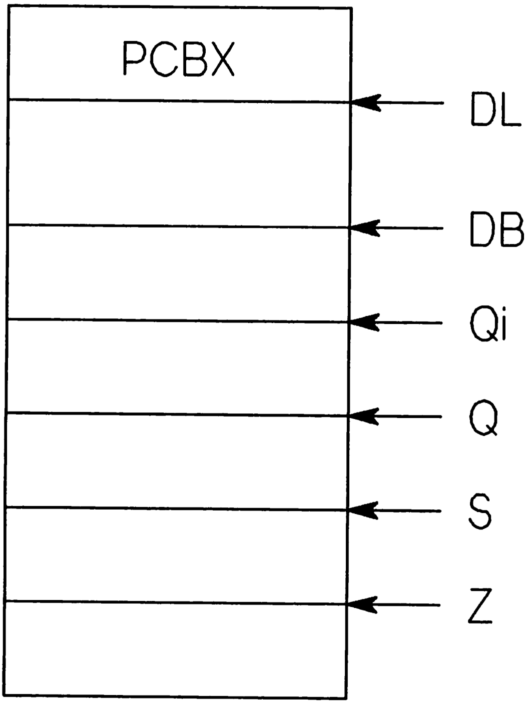


Figure 2a

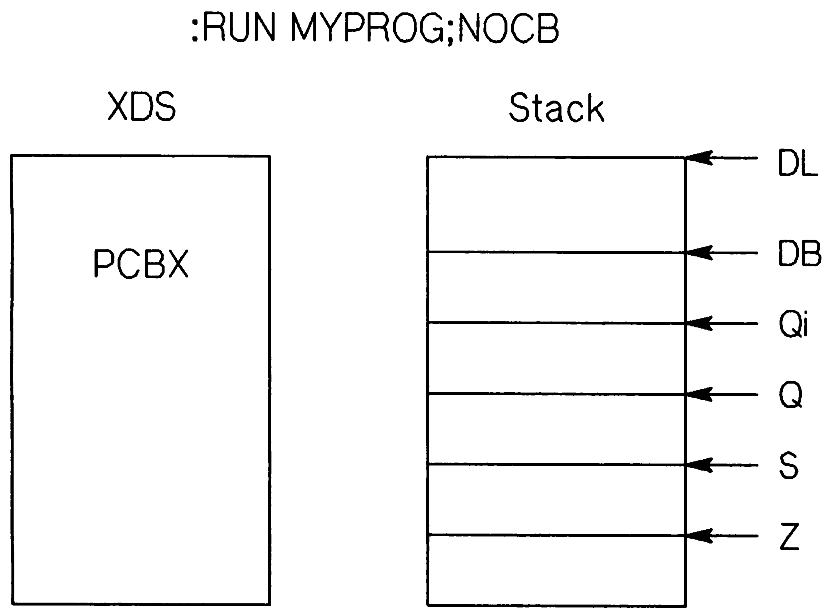
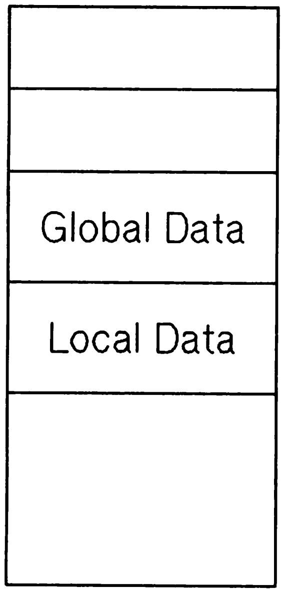


Figure 2b

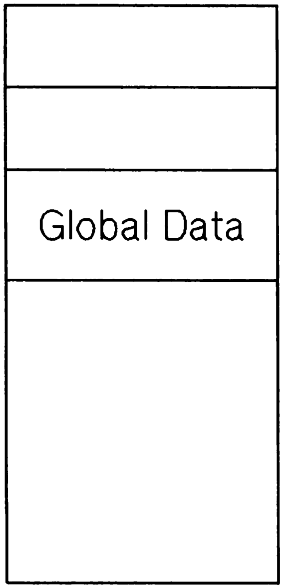
Static Subprogram



Stack after GOBACK from subprogram

Figure 3

Dynamic Subprogram



Stack after GOBACK from subprogram

Figure 4

Nesting Subprograms

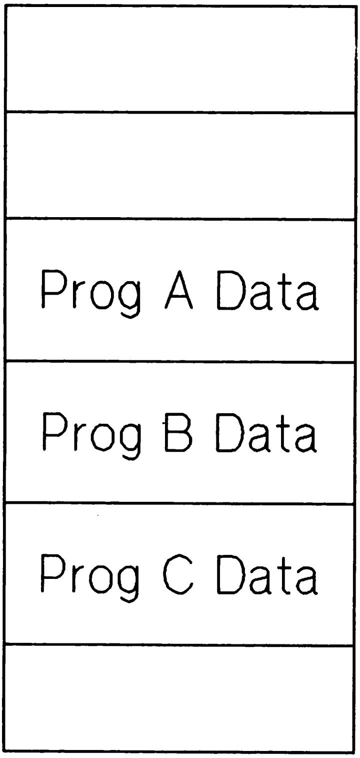


Figure 5

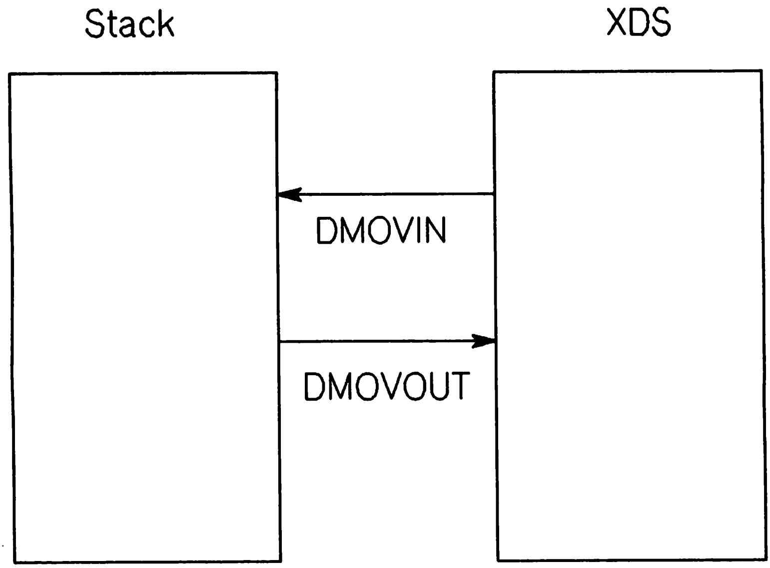


Figure 6

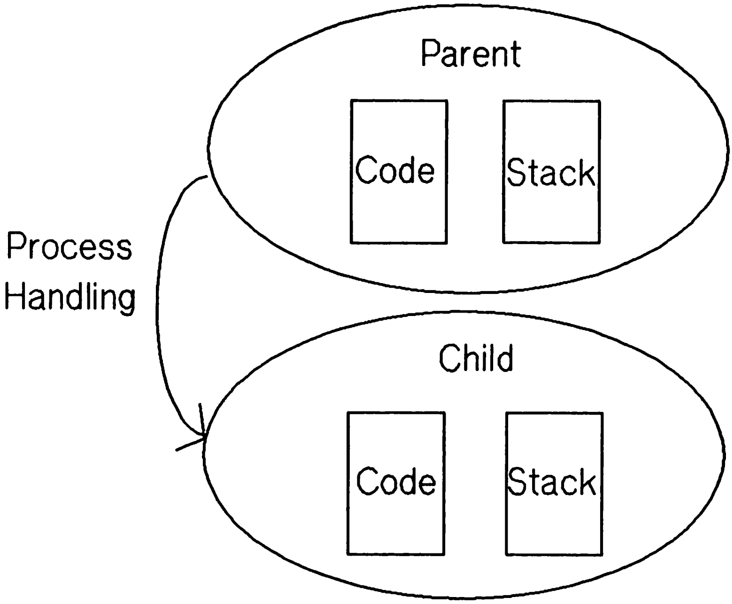


Figure 7

Figure 8

Process display within Process context of OPT.PUB.SYS

```
PIN: 403  MYPROG.MYGROUP.MYACCT        USER:  ME.MYACCT      (S227)  1:12 PM
-------------------------------------------------------------------------------
        STACK INFORMATION         | CPU TIME:   7057 MSEC|STATUS FLAGS:
    DST:  1403* SYSOV:  1329  4.8%|                      |
   SIZE: 27648  DL-DB:  7640 27.6%| PRIORITY: 152        |MRNG  GRIN  LRIN
 MAXDATA: 31223  DB-QI: 10967 39.7%| CAPABILITIES:   ND SF|BIO   I/O   UCOP
MAX Z-DL: 22955   QI-Q:  3924 14.2%|  BA IA PH DS         |JUNK  TIME  MSG
                  Q-S:     9   .0%|                      |
                  S-Z:  3779 13.7%|                      |
-------------------------------------------------------------------------------
```

A version of this article originally appeared in the Nov. 1987 edition of HP Professional magazine.

Giving Them What They Want: Quick Prototyping on the HP3000

Lisa Burns Hartman
Hewlett-Packard
Corporate Offices
P. O. Box 10301
Palo Alto, CA 94303

If your experience is anything like ours, you have had mixed success with various methods of specifying systems. Specifying new systems is a difficult process, and can result in lengthy documents which are never read. Even for enhancements to existing systems, it can be difficult to express on the printed page what the new feature will do. Using diagrams of new screens can help, but it is not the same as logging on and interacting with a system. Users who receive systems which were designed only with written documents may be surprised and unhappy when the system arrives.

Prototyping

Our first answer to this problem was to build prototypes of new features. This was more successful than a written document for several reasons. Our users could see the screens online. This allowed them to try out the ergonomics of the screens: the video enhancements, the softkeys, the TABing, the placement and length of the fields, and the overall aesthetics of the screen itself. By interacting with a working prototype they could type in transactions, see data returned to them and clear up any confusion about how the new screen or feature was really going to work. This was infinitely more informative than reading a document.

However, building the prototypes was not cheap. We did not want to use a Fourth Generation Language because of the heavy interaction with our existing system for some of these prototypes. It was difficult to integrate prototypes written in a 4GL with existing programs in COBOL. So, since we used COBOL for the prototypes, it took 6 weeks or more to develop some of them. By that time, our investment in the design as we saw it was quite high. However, our prototype sessions proved that our idea of for a new feature did not always work in a real office environment. If users wanted significant changes to the initial design, it would be quite expensive to develop another prototype. It was very tempting to go ahead and install the code we had already developed and to ignore the users' input! We were not catching design problems or possible design improvements early enough in the process.

Prototyping using V/PLUS and ENTRY

Because of these experiences, we decided to try another approach. By using the FORMSPEC subsystem within V/PLUS, we designed screens for our new features. By linking these screens together and adding plausible data to be displayed to the user, we built a simulated prototype of the new feature which could be run using the ENTRY subsystem of V/PLUS. This prototype included a narrative script where users are told to enter values in the various fields to set up a transaction. The entire setup for the prototype took our summer student, who was not familiar with FORMSPEC, less than two days. Not a single line of code needed to be written!

Advantages of ENTRY prototyping

When we sat down with users to show them the new screens, they were very pleased. As with traditional prototyping, they could see the screens online. They could review the placement of fields, which fields they wanted to see, the video enhancements to be used, and the overall appearance of the screen. They could enter transactions according to instructions in our narrative and work with TABing and length of fields. Again, this was much better than asking them to respond to a written document, and much more informative.

You may have already used ENTRY to allow users to try out screen designs. It is an excellent way to get feedback quickly. But blank screens have limitations. Without displaying data back to the users, we cannot meet the goal of simulating an interactive program. However, by linking several copies of the same screen together, and placing data in the fields in the second screen, we can simulate a program interacting with the user. The users can respond to the demo as though there was a program returning the data, and give us feedback on the way the function was designed.

How to set up an ENTRY prototype

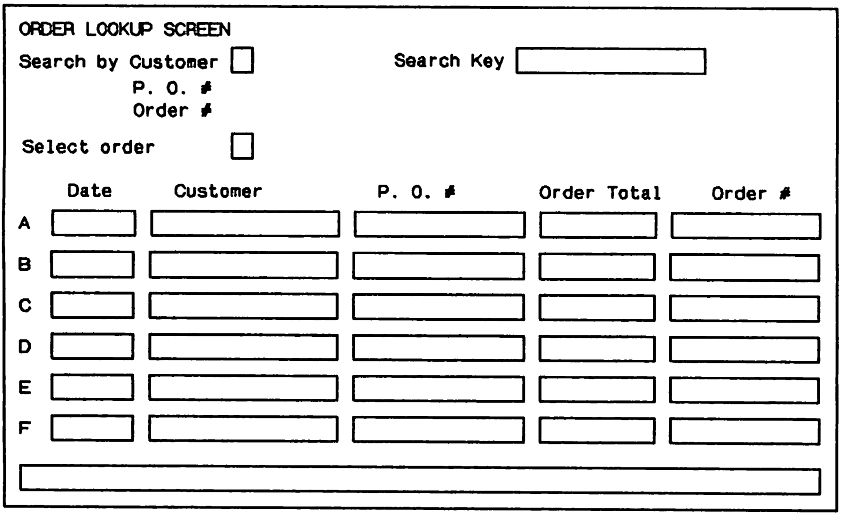
To illustrate how to set up such a simulation, let's take an example of an order inquiry screen. Suppose that we are going to design an inquiry screen to access an order database. The first step is to run FORMSPEC.PUB.SYS, and set up the new screen. Screen 1 shows the design of our first screen when brought up in ENTRY. This will be the first screen the users see when they run the prototype.

Now that we have our basic screen design, let's think about how our prototype will work. With this screen, the user will be able to query our database using three different keys: Customer, P. O. number, and Order number. How might a user expect this screen to operate? How will the searches work? Will partial keys be allowed? How will the orders be sorted when they do appear? Our prototype sessions with the users will answer these questions.

We need to set up demonstrations to answer each of our questions. First of all, let's set up an example for customer searching. First, we make a copy of the screen to a second screen name in FORMSPEC. Then, we link these two screens together using the NEXT FORM field. Then, working on the second screen, we use the Initial Values field in FORMSPEC to set up prototype data. I found that printing the sample screen and writing in plausible values for each field made the data entry of initial values in FORMSPEC much easier. The results of this data entry are shown in Screen 2.

In only a few minutes we have just written a small prototype! By running ENTRY.PUB.SYS and specifying the FORMSFILE name, we can run the prototype. The user first sees the blank basic screen (Screen 3a). Our narrative calls for a demonstration of the customer search feature. In our script, the user is asked to type a "C" for Customer search in the first field of the screen, and then to type the customer name "ABC CORPORATION" (Screen 3b). The user then hits <ENTER>, and the second screen we created is painted, with our sample data from the Initial Values fields in FORMSPEC (Screen 3c).

SCREEN 1


ORDER LOOKUP SCREEN
Search by Customer
P. O. #
Order #
Search Key
Select order
Date
Customer
P. O. #
Order Total
Order #
A
B
C
D
E
F

SCREEN 2

ORDER LOOKUP SCREEN

Search by Customer [C]   Search Key [ABC CORPORATION]

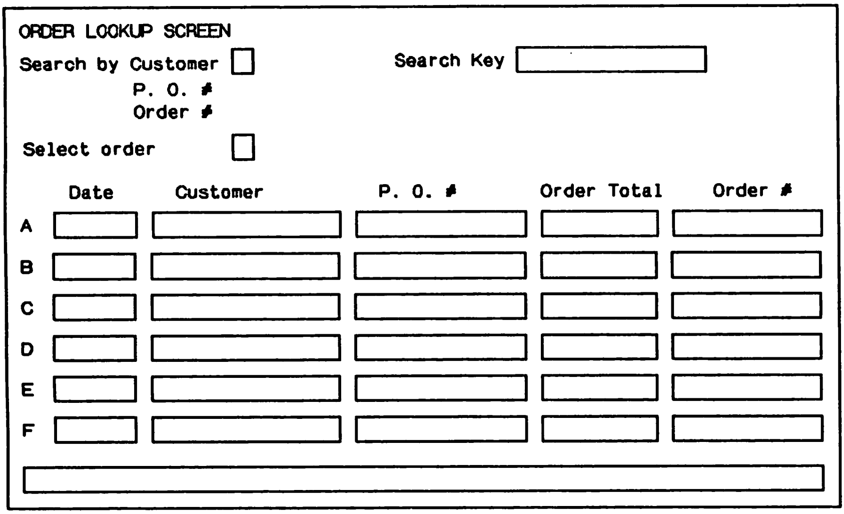
P. O. #

Order #

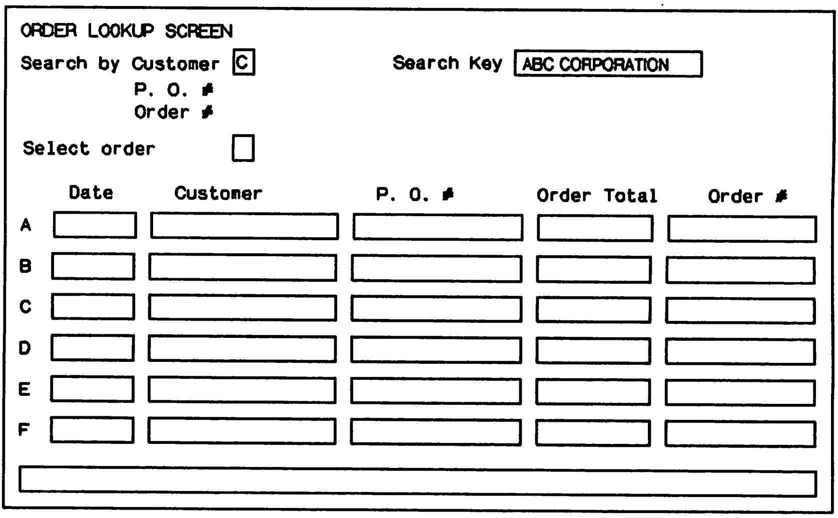
Select order [ ]

	Date	Customer	P. O. #	Order Total	Order #
A	061587	ABC CORPORATION	1234567	$ 5100.00	ABCD12345678
B	062587	ABC CORPORATION	1234567	$ 900.00	ABCD12345679
C	062587	ABC CORPORATION	1234567	$ 7500.00	ABCD12345679
D	061587	ABC CORPORATION	1234567	$ 8800.00	ACDF99887766
E	063087	ABC CORPORATION	1234567	$ 100.00	ACDF77885544
F					

SCREEN 3A


ORDER LOOKUP SCREEN
Search by Customer
P. O. #
Order #
Search Key
Select order
Date
Customer
P. O. #
Order Total
Order #
A
B
C
D
E
F

SCREEN 3B


ORDER LOOKUP SCREEN
Search by Customer
C
P. O. #
Order #
Search Key
ABC CORPORATION
Select order
Date
Customer
P. O. #
Order Total
Order #
A
B
C
D
E
F

SCREEN 3C

ORDER LOOKUP SCREEN

Search by Customer [C] Search Key [ABC CORPORATION]

P. O. #

Order #

Select order [ ]

	Date	Customer	P. O. #	Order Total	Order #
A	061587	ABC CORPORATION	1234567	$ 5100.00	ABCD12345678
B	062587	ABC CORPORATION	1234567	$ 900.00	ABCD12345679
C	062587	ABC CORPORATION	1234567	$ 7500.00	ABCD12345679
D	061587	ABC CORPORATION	1234567	$ 8800.00	ACDF99887766
E	063087	ABC CORPORATION	1234567	$ 100.00	ACDF77885544
F					

This example illustrates the basic concept we followed in creating our prototype: Show the basic screen, tell users what to enter in each field, then show the next screen filled with data corresponding to the entered data.

The same technique can be used to illustrate alternatives for implementing a particular program function. Screens 4a-d illustrate a data inquiry function which could display data in one of three different sort orders. Users viewing this prototype could choose the sort order which most closely fits their needs.

What else can an ENTRY prototype show our users? By making copies of the basic screen and then varying the copies, we can demonstrate the effect of different video enhancements. Screen 5 shows a variation of our basic screen with underline enhancements for some of the fields rather than half-inverse. ENTRY can also allow users to experiment with field placement. Screen 6 and Screen 7 show the same data elements arranged differently on each screen. By having users enter data from actual transactions into these screens, they can determine the best placement of the fields for ease of data entry. The placement of fields and the video enhancements used may seem trivial to a programmer, but to someone who stares at screens eight hours a day, these are very important issues.

By linking various scenarios together, we can set up a prototype session illustrating several alternatives for a new screen design. This session will yield very effective user feedback and help us resolve design issues before we write a single line of code.

SCREEN 4A

ORDER LOOKUP SCREEN

Search by Customer [P]   Search Key [9900]
P. O. #
Order #

Select order [ ]

	Date	Customer	P. O. #	Order Total	Order #
A					
B					
C					
D					
E					
F					

SCREEN 4B -- P. O. # sort

ORDER LOOKUP SCREEN

Search by Customer [P]   Search Key [9900]
P. O. #
Order #

Select order [ ]

	Date	Customer	P. O. #	Order Total	Order #
A	062587	XYZ CORP	990075	$ 7790.00	ABCD00000001
B	062787	XYZ CORP	990076	$ 8876.00	ABCD00000017
C	060187	XYZ CORP	990077	$ 9950.00	ABCD00000005
D	060587	XYZ CORP	990078	$ 60.00	ABCD00000007
E	071087	XYZ CORP	990079	$ 1500.00	ABCD00000003
F					

SCREEN 4C -- Order # sort

ORDER LOOKUP SCREEN

Search by Customer P   Search Key 9900
P. O. #
Order #

Select order

	Date	Customer	P. O. #	Order Total	Order #
A	062587	XYZ CORP	990075	$ 7790.00	ABCD00000001
B	071087	XYZ CORP	990079	$ 1500.00	ABCD00000003
C	060187	XYZ CORP	990077	$ 9950.00	ABCD00000005
D	060587	XYZ CORP	990078	$ 60.00	ABCD00000007
E	062787	XYZ CORP	990076	$ 8876.00	ABCD00000017
F					

SCREEN 4D -- Date sort

ORDER LOOKUP SCREEN

Search by Customer [P] Search Key [9900]
P. O. #
Order #

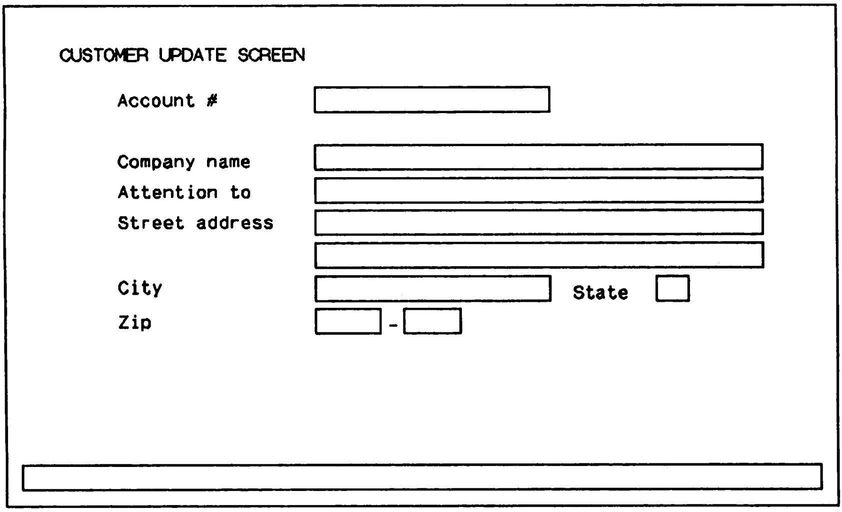
Select order [ ]

	Date	Customer	P. O. #	Order Total	Order #
A	060187	XYZ CORP	990077	$ 9950.00	ABCD00000005
B	060587	XYZ CORP	990078	$ 60.00	ABCD00000007
C	062587	XYZ CORP	990075	$ 7790.00	ABCD00000001
D	062787	XYZ CORP	990076	$ 8876.00	ABCD00000017
E	071087	XYZ CORP	990079	$ 1500.00	ABCD00000003
F					

2068-15

2068-16

SCREEN 7


CUSTOMER UPDATE SCREEN
Account #
Company name
Attention to
Street address
City
State
Zip

The prototype session

Writing the prototype is only half the battle for getting user input. By asking the right questions, we can elicit very important information from our users. The following are some tips we have found helpful when setting up an ENTRY prototype session.

- Schedule the prototype meeting at a time when users will have an hour or so to devote to this activity. We have found that our users are very excited to participate in the sessions, so getting them to take time to see the prototype is not usually a problem.

- Set up the prototype in a room where the users and the designers will not be distracted. This allows everyone to concentrate on the session.

- Set up several terminals, one for every 2-3 people. This allows each person to get hands-on experience with the screens.

- Give each user a copy of your narrative so that they understand the test data to be entered. Include instructions for running ENTRY.PUB.SYS and putting up the first screen. This will minimize confusion during the session and allow easy data entry according to the narrative.

Now it's time to begin the session. As each screen comes up, ask the users questions: Do you like the order of the fields? Do you like the video enhancements? Is the screen too busy? Are the field labels clear? and so on. Have them tab around the screen and enter sample transactions. Ask them if the fields are in an intuitive order, if seldom-entered fields are at the end, and if tabbing works the way they would like.

As you display each alternative design, ask which features they like the most: Does this sort order work best for you? Do you need a search on less than a full key? Is enough data displayed? Are error messages self explanatory? Does the system respond in a way which makes sense to you? Does it operate in a manner which fits the way you do business? You may need to ask leading questions in order to stimulate discussion.

You may wish to designate one of the designers as a scribe so that answers to these questions can be recorded. When the session is over, the scribe should summarize the results of the session, noting users' preferences and any changes to the screens which were suggested. As a final check, you may wish to run a second session with the combined results in a new screen or screens. This will help ensure that you heard your users correctly.

Once your prototype sessions are completed, you should publish a document with printed screen layouts explaining your final external specification. Hopefully, this document will be far shorter than one generated without user input, since design alternatives have been eliminated and issues have been resolved. You can then proceed to construct the new screens or system, confident that you are creating something that your users will like and which will fit their needs.

Limitations and advantages

There are a few minor limitations of ENTRY prototyping. Since there is no application program running behind the screens, response time will not be accurate. Also, softkey functionality cannot be demonstrated for menu-driven applications. Finally, the cursor will always return to the first field on a given form. For applications doing cursor placement in a field somewhere else on the screen, this may confuse users.

The advantages of ENTRY prototyping outweigh the limitations for many applications. The technique allows users to interact with the system and provides most of the benefits that a prototype program provides. It is not necessary for designers to learn a new language. If their application is V/PLUS based they are already familiar with FORMSPEC. But by far the biggest advantage of ENTRY prototyping is the speed of development. I developed all of the screens for this article in under two hours. Modifications to screens can also be made very quickly.

Conclusion

Using FORMSPEC and ENTRY to set up prototypes can be very valuable for application teams. A day or two invested up front can save weeks and months of redesign later in the development cycle. ENTRY prototyping lets the user interact with the proposed system hands-on, on-line. By showing users exactly what the system will do and letting them see and touch a prototype, developers can be confident that the product they will deliver is what the user wants. Our experience with ENTRY prototyping has been extremely positive. We plan to expand its use for new enhancements to our systems.

A version of this article originally appeared in the October, 1987 of HP Professional magazine.

Prototyping on the HP3000

2068-20

What's in HP Pascal: A Systems Programming Language

Sue Kimura

Hewlett-Packard Company
19447 Pruneridge Avenue
Cupertino, California 95014

Introduction

HP Pascal has been enhanced to include features which allow it to be a systems programming language. These features have made it possible to write the MPE/XL operating system in HP Pascal. The HP Pascal compilers are available on Hewlett-Packard Precision Architecture (HPPA) systems. 1

Historically, Pascal has had a reputation as a student's language. It is known for its structured constructs and strict typing rules. There is no doubt that its structured constructs make it attractive as a programming language. Its strict typing rules, however, while helping the programmer avoid run-time problems, have made it difficult for it to be used as a systems programming language.

This paper focuses on the following systems language features:

```
New Data Representation
Type Coercion
Generic Pointers
Procedure and Function Extensions
Dynamic Routines
Exception Handling
Move Routines
Building Intrinsic Files
```

While descriptions and examples of these features are given, this paper is not a tutorial. The *HP Pascal Reference Manual* and *HP Pascal Programmer's Guide* are available for complete explanations of these features.

To permit access to these system language features, either of the compiler options `standard_level 'hp_modcal'` or `standard_level 'ext_modcal'` is required.

1HP Pascal is a superset of the ANSI/IEEE770X3.97-1983 and ISO 7185:1983 standards.

New Data Representation

A short integer data type, generic pointer types, procedure and function types, and a crunched attribute have been added to HP Pascal.

Shortint

A predefined, short integer data type is available in HP Pascal. `Shortint` is a 16-bit, 2-byte aligned data type. Note that shortint is not the same as the subrange -32768..32767 which in HP Pascal is a 32-bit, 4 byte-aligned data type.

Its purpose is to handle compatibility with the MPE/V operating system. It is analogous to the SPL/V integer data type.

The shortint data type does not require the compiler option `standard_level 'hp_modcal'` or `standard_level 'ext_modcal'`.

Localanyptr, Globalanyptr, Anyptr

Another set of new data types are the generic pointer types: `localanyptr`, `globalanyptr`, and `anyptr`. We will discuss these pointer types later, under the topic **Generic Pointers**.

Procedure and Function Types

Procedure and function types are used to define routines which are dynamically invoked at run-time. We will discuss these types later, under the topic **Dynamic Routines**.

Crunched Structures

In addition to allowing packed structures, HP Pascal allows bit packing of data with crunched structures. In this form of data representation no bits are wasted. This allows the programmer to have the greatest control in determining the layout of data.

The crunched attribute in a structure declaration overrides the alignment restriction for allowed types. The allowed types are integer, shortint, boolean, char, enumeration, and subrange of integer, boolean, char, and enumeration. Crunched structures (e.g., array, record, set) of these types are also allowed.

For example, an integer is 4-byte aligned in an unpacked or packed record. In a crunched record it is bit-aligned.

Note the difference in the data representation of the following records, which are unpacked, packed and crunched:

```
unpacked_record = RECORD
    f1 : 0..7;              {1 byte,  1-byte aligned}
    f2 : 0..255;            {1 byte,  1-byte aligned}
    f3 : 0..65535;          {2 bytes, 2-byte aligned}
    f4 : -32768..32767;     {4 bytes, 4-byte aligned}
    f5 : shortint;          {2 bytes, 2-byte aligned}
    f6 : integer;           {4 bytes, 4-byte aligned}
    END;
       {total size = 16 bytes, record alignment = 4-bytes}

 packed_record = PACKED RECORD

    f1 : 0..7;              {3 bits,  1-bit aligned}
    f2 : 0..255;            {1 byte,  1-bit aligned}
    f3 : 0..65535;          {2 bytes, 1-bit aligned}
    f4 : -32768..32767;     {2 bytes, 1-bit aligned}
    f5 : shortint;          {2 bytes, 2-byte aligned}
    f6 : integer;           {4 bytes, 4-byte aligned}
    END;
       {total size  = 12 bytes, record alignment = 4-bytes}

 crunched_record = CRUNCHED RECORD
    f1 : 0..7;              {3 bits,  1-bit aligned}
    f2 : 0..255;            {1 byte,  1-bit aligned}
    f3 : 0..65535;          {2 bytes, 1-bit aligned}
    f4 : -32768..32767;     {2 bytes, 1-bit aligned}
    f5 : shortint;          {2 bytes, 1-bit aligned}
    f6 : integer;           {4 bytes, 1-bit aligned}
    END;
       {total size = 91 bits, record alignment = 1-bit}
```

A crunched record is most useful when the programmer needs to control the layout of data. For example, he may need to copy the data layout of other machines. However, accessing data when they are not aligned on byte-boundaries is costly. Obviously, it is a space over performance tradeoff.

Type Coercion

Type coercion is a mechanism for circumventing the strict typing rules of Pascal. It is enabled by the compiler option type_coercion.

Type coercion allows one type of data to be represented as another type. The type of the expression being coerced is called the source type, and the type the expression is being coerced to is called the target type.

The syntax of type coercion is identical to that of a function call:

```
target_type (source_expression)
```

Note the term source_expression. This term indicates that type coercion may not be used on the left-hand side of an assignment statement.

There are five levels of type coercion. In order of decreasing restrictiveness these levels are:

    conversion
    structural
    representation
    storage
    noncompatible

Conversion type coercion is of two types: ordinal type conversion and pointer type conversion.

Ordinal type conversion is used to convert an ordinal type (integer, shortint, enumeration, boolean, char, subrange) to another ordinal type. It is most useful when converting from an enumerated type to an integer type and vice versa. Range checking is done to insure that the value of the source expression is within the range of the target type.

Example

```
$standard_level 'ext_modcal'$
$type_coercion 'conversion'$
PROGRAM ordinal_type_coercion;
TYPE
   spectrum = (red, orange, yellow, green, blue, violet);
VAR
   rainbow : spectrum;
   i       : integer;
BEGIN
rainbow := orange;
i := integer (rainbow);    {i := 1}
i := i + 1;
rainbow := spectrum (i);   {rainbow := yellow}
END.
```

Pointer type conversion is used to change from one pointer type to another pointer type. It may be a short-to-short, short-to-long, long-to-long, or long-to-short pointer conversion. Long-to-short pointer conversion may cause a run-time range error. We will discuss short and long pointers later, under the topic **Generic Pointers**.

The remaining levels of coercion may be viewed as the overlaying of storage of tagless variants within a record. This form of coercion is also called free union coercion. Unlike conversion type coercion, no range checking is done.

The differences in these levels are based on the restrictions regarding the storage allocated for the source and target types, their alignment and their type compatibility (Table 1). The specific rules for type compatibility are described in the *HP Reference Manual*.

Level of Coercion	Storage	Alignment	Type Compatibility
Structural	S = T	S = T	Compatible
Representation	S = T	NR	NR
Storage	S >= T	NR	NR
Noncompatible	NR	NR	NR

S = Source Type
T = Target Type
NR = No Restriction

Table 1. *Restrictions for Free Union Coercion*

Structural type coercion, the most restrictive form of free union coercion, requires that the storage and alignment of the source and target type be the same. Their types must also be compatible. If the source and target types are structures, their component types must also follow these rules.

Example

```
$standard_level 'ext_modcal'$
$type_coercion 'structural'$
PROGRAM structural_type_coercion;
TYPE
   arrtype1 = ARRAY [1..10] OF integer;
   arrtype2 = ARRAY [1..10] OF minint..maxint;
CONST
   ca2 = arrtype2 [10 OF -1000];
VAR
   a1 : arrtype1;
   a2 : arrtype2;
BEGIN
a2 := ca2;              {a2[1] := -1000, ..., a2[10] := -1000}
a1 := arrtype1 (a2);    {a1[1] := -1000, ..., a1[10] := -1000}
END.
```

Structural type coercion is used to assign a2 to a1. It is allowed because both a1 and a2 are arrays with elements that have the same size (32 bits), have the same alignment (4-byte), and are type compatible. The result of the assignment is that each element of a1 has the value -1000.

The remaining three levels do not have alignment and type restrictions. Representation type coercion requires the same storage for the source and target types. Storage type coercion requires the target type to be the same size or smaller than the source type.

Both representation and storage type coercion guarantee that no undefined bits in the target type are accessed. However, undefined bits in the source type may still be accessed. This may occur if the source type has undefined bits because of its packing.

Example

```
$standard_level 'ext_modcal'$
PROGRAM representation_and_storage_type_coercion;
TYPE
   rectype1 = RECORD
      f1 : integer;     {4 bytes, 4-byte aligned}
      END;
         {total size = 4 bytes, alignment = 4 bytes}

   rectype2 = RECORD
      f1 : shortint;    {2 bytes, 2-byte aligned}
      f2 : shortint;    {2 bytes, 2-byte aligned}
      END;
         {total size = 4 bytes, alignment = 2-bytes}

   rectype3 = RECORD
      f1 : boolean;     {1 byte,  1-byte aligned}
      f2 : boolean;     {1 byte,  1-byte aligned}
      f3 : shortint;    {2 byte,  2-byte aligned}
      f4 : shortint;    {2 byte,  2-byte aligned}
      END;
         {total size = 6 bytes, alignment = 2-bytes}
CONST
   cr2 = rectype2 [f1: 0, f2: 1];
   cr3 = rectype3 [f1: false, f2: false, f3: 3, f4: -32768];
VAR
   r1 : rectype1;
   r2 : rectype2;
   r3 : rectype3;
BEGIN
r2 := cr2;                            {initialize r2}
r3 := cr3;                            {initialize r3}
$type_coercion 'representation'$
r1 := rectype1 (r2);                  {r1.f1 := 1}
$type_coercion 'storage'$
r1 := rectype1 (r3);                  {r1.f1 := 3}
END.
```

Representation type coercion is used to assign r2 to r1. Both r1 and r2 are records and take the same amount of storage. Note, however, that r1 and r2 do not have the same alignment; r1 is 4-byte aligned while r2 is 2-byte aligned. The result of the assignment is that r1.f1 has the value 1.

Storage type coercion is used to assign r3 to r1. R3 is larger than r1; consequently, any bits not defined for the type rectype1 is not accessible to r1. Specifically, r3.f4 is not accessible to r1. The result of the assignment is that r1.f1 has the value 3.

Finally, noncompatible type coercion allows any type to be coerced to any other type. As the least restrictive form, it is the most dangerous to use.

Example

```
$standard_level 'ext_modcal'$
PROGRAM noncompatible_type_coercion;
TYPE
   rectype1 = RECORD
      f1 : integer;     {4-bytes, 4-byte aligned}
      END;
   rectype4 = RECORD
      f1 : boolean;     {1-byte, 1-byte aligned}
      END;
VAR
   r1 : rectype1;
   r4 : rectype4;
BEGIN
r4.f1 := false;
$type_coercion 'noncompatible'$
r1 := rectype1 (r4);                {r1.f1 := ??}
END.
```

Noncompatible type coercion is used to assign r4 to r1. Because r4 is smaller than r1, r1.f1 accesses bits not defined for r4. The result of the assignment is a garbage value in r1.f1.

As shown in the above examples, the general rule when using type coercion is obvious: use the most restrictive form of coercion that gets the job done.

Note also that type coercion is applicable at the statement level. Only statements that need type coercion should be bracketed with the appropriate level. A common method used to bracket a type coercion statement is to use the compiler options push and pop:

```
$push, type_coercion 'representation'$
r1 := rectype1 (r3);
$pop$
```

Generic Pointers

Generic pointers are different from the typed pointers in Pascal which manipulate the heap. They are true addresses.

There are two types of generic pointers on a HPPA system. A `long pointer` can point to any addressable object on a HPPA system. A `short pointer` points to a subset of these addressable objects.

HP Pascal defines three pointer types: `localanyptr`, `globalanyptr` and `anyptr`. `Localanypointer` is a 32-bit or short pointer. `Globalanyptr` is a 64-bit or long pointer. `Anyptr` on a HPPA system is a globalanyptr. Since the definition of anyptr may change from system to system, it is wise to use `localanyptr` if a short pointer is desired, or `globalanyptr` if a long pointer is desired.

A long pointer is created using the compiler option `extnaddr` in a type, variable, or formal parameter declaration. Long pointers are primarily used by the operating system and subsystems. Users do not normally need to use long pointers.

Generic pointers are assignment compatible with any other pointer type. Their primary restriction is that they may not be dereferenced. In other words, to access data, a generic pointer must be assigned or coerced to a typed pointer.

Two predefined routines allow the manipulation of these pointers. The predefined function `addr` creates a reference to data. The address returned may point to data in the heap, or to local or global data.

The predefined function `addtopointer` allows for arithmetic manipulation of an address. `Addtopointer` returns a pointer value that is a programmer-specified number of bytes away from the current pointer value.

The preferred way to perform address manipulation is to use these generic pointers and predefined routines, rather than to use tagless variant records. Using these routines allows the HP Pascal compiler to generate more optimal code.

The following is an example of walking through an array and printing out contents of its elements:

Example

```
$standard_level 'ext_modcal'$
PROGRAM generic_pointer (output);
TYPE
   intarrtype = ARRAY [1..20] of integer;
   iptrtype =  ^integer;
CONST
   cintarr = intarrtype [20 of 0];
VAR
   intarr : intarrtype;
   ptr,
   endptr : localanyptr;
   iptr   : iptrtype;
BEGIN
   {initialize elements of intarr to 0}
intarr := cintarr;
   {determine the starting address of intarr}
ptr := addr (intarr);
   {determine the ending address of intarr}
endptr := addtopointer (ptr, sizeof (intarr));
WHILE ptr <> endptr DO
   BEGIN  {print next element}
   iptr := ptr;
   writeln (iptr^);
   ptr := addtopointer (ptr, sizeof (integer));
   END;  {print next element}
END.
```

In this example, addr is used to set the base address of the array intarr. Addtopointer is used to determine the ending address as well as to determine the address of the next element of intarr. Sizeof is used to obtain the size in bytes of the array intarr and of the type integer. Because ptr is a localanyptr it cannot be dereferenced to access the data in intarr. Consequently, ptr is assigned to a typed pointer, iptr, and iptr is dereferenced.

Procedure and Function Extensions

New features have been added to the mechanism for declaring a routine and its parameters. These include an anyvar reference parameter, and options for providing default values for parameters, for making parameters extensible, and for duplicating routine code.

ANYVAR

A formal parameter may be declared as ANYVAR. An anyvar parameter is a reference parameter that accepts an actual parameter of any type. The data that are passed are treated as the type of the formal parameter. In other words, ANYVAR is a form of noncompatible type coercion.

When a parameter is declared as ANYVAR a byte count representing the size of the actual parameter is also passed along with the address of the actual parameter. This information may be used to insure that storage allocated for the actual parameter is not overwritten, as well as to refrain from accessing undefined storage in the actual parameter. The byte count may only be accessed by calling the predefined function sizeof.

The following is an example of copying data from one array to another using an ANYVAR parameter and a VAR parameter.

Example

```
$standard_level 'hp_modcal'$
PROGRAM anyvar_parm;
TYPE
   spac = PACKED ARRAY [1..10] OF char;
   lpac = PACKED ARRAY [1..20] OF char;

VAR
   i : integer;
   sp : spac;
   lp : lpac;

PROCEDURE copy_data (
   ANYVAR fromparm : spac;
   VAR    toparm : spac);
VAR
   i : integer;
BEGIN
i := 1;
WHILE (i <= sizeof (fromparm))
      AND
      (i <= sizeof (toparm)) DO
   BEGIN
   toparm[i] := fromparm [i];
   i := i + 1;
   END;
END;

BEGIN
...
copy_data (i, sp);       {first call}
copy_data (lp, sp);      {second call}
END.
```

In this example, the difference between fromparm and toparm is that a variable of any type may be passed to fromparm, but only a variable of type spac may be passed to toparm. Sizeof(fromparm) is called to insure that only the data defined for fromparm is assigned to toparm, as in the first call to copy_data. Sizeof(toparm) is called to insure that toparm does not go beyond its bounds in the case that fromparm is larger than toparm, as in the second call to copy_data.

Uncheckable_anyvar

If the byte count of an anyvar parameter is not needed or desired, the anyvar parameter should be declared as OPTION uncheckable_anyvar. In this case the sizeof function returns the size of the formal parameter, rather than the size of the actual parameter.

Example

```
PROCEDURE copy_data (
   ANYVAR fromparm : spac;
          fromparmlen : integer;
   VAR    toparm : spac )
   OPTION uncheckable_anyvar;
   external c;
```

In this example, OPTION uncheckable_anyvar is used to eliminate the byte count and the caller is responsible for passing the size of fromparm.

OPTION uncheckable_anyvar should be used when declaring non-Pascal routines which do not support ANYVAR, or when declaring Pascal routines which are to be called from non-Pascal routines.

Default Parameters

Initialization of parameters is provided by declaring default parameters. Default parameters allow empty actual parameters to be passed.

Default parameters are declared with OPTION default_parms following a routine parameter list. A value is required for each of the defaulted parameters. A reference parameter is only allowed the default value nil.

In a routine with default parameters, the predefined function haveoptvarparm may be used for a formal reference parameter to determine whether an actual parameter was defaulted or supplied by the caller. For a formal value parameter, there is no way to determine whether an actual parameter was defaulted or supplied.

The following is an example of opening a Pascal textfile using default parameters for the name of the file to be opened and length of the file name.

Example

```
$standard_level 'ext_modcal'$
PROGRAM default_parms;
CONST
   maxlen = 1024;
TYPE
   lenrange = 0..maxlen;
   pac = PACKED ARRAY [1..maxlen] OF char;
VAR
   pacv : pac;
   f    : text;
PROCEDURE open_file (
   VAR  f        : text;
   VAR  filename : pac;
        length   : lenrange
   ) OPTION default_parms (filename := nil,
                           length := 0
                          );
VAR
   i : integer;
   fname : PACKED ARRAY [1..maxlen+1] OF char;
BEGIN
IF (haveoptvarparm (filename))
   AND
   (length > 0) THEN
   BEGIN  {file name has been passed}
   FOR i := 1 TO length DO
     fname[i] := filename[i];
   fname[length+1] := ' ';
   rewrite (f, fname);
   END  {file name has been passed}
ELSE
   rewrite (f, '$stdlist');
END;

BEGIN
pacv := 'xxxxx';
open_file (f,pacv, 5);    {first call}
open_file (f);            {second call}
END.
```

In the above example, the first parameter, f, must be passed because it does not have a default value. The remaining two parameters, filename and length, have default values and do not need be passed by the caller.

The first call to open_file opens the file called 'xxxxx'. The second call opens the standard output file because no parameters were passed for filename and length. The predefined function haveoptvarparm is called to determine if an actual parameter was passed, as in the first call, or defaulted, as in the second call. `Length` is checked to verify that it is at least 1.

Extensible Parameters

A routine may also have extensible parameters. Extensible parameters are those which are not required at the end of a parameter list when the routine is called.

Extensible parameters are declared with OPTION `extensible` *n* following a routine parameter list. The value *n* indicates that the first *n* parameters are required. In other words, these are the non-extensible parameters. The value of *n* may be between 0 and the number of parameters declared for the routine.

In the extensible routine, the predefined function `haveextension` may be used to determine if an extensible parameter has been passed.

The following is also an example of opening a Pascal textfile. In this instance, however, the `extensible`, rather than the `default_parms` option is used.

Example

```
PROGRAM extensible_parameters;
...
PROCEDURE open_file (
   VAR  f        : text;
   VAR  filename : pac;
        length   : len )
   OPTION extensible 1;
VAR
   i : integer;
   fname : PACKED ARRAY [1..maxlen+1] OF char;
BEGIN
IF (haveextension (filename))
   AND
   (haveextension (length)) THEN
   BEGIN
   FOR i := 1 TO length DO
     fname[i] := filename[i];
   fname[length+1] := ' ';
   rewrite (f, fname);
   END
ELSE
   rewrite (f, '$stdlist');
END;

BEGIN
pacv := 'xxxxx';
open_file (f,pacv, 5);    {first call}
open_file (f);            {second call}
END.
```

In this example, the first parameter is non-extensible and must be passed by the caller. The remaining two are extensible and do not need to be passed. In the procedure `open_file` the predefine `haveextension` is called to determine if the extensible parameters have been passed.

Note that the calls to `open_file` are identical to those in the default parameters example.

The `extensible` and `default_parms` options may be used together. The semantics of combining these options are described in the *HP Pascal Programmer's Guide.*

Inlining Routines

Sometimes it is useful to have routines that have very simple bodies, such as routines to push and pop items from a stack. The programmer has a choice of calling a procedure or duplicating the same code in each place it is needed. A procedure call may be more readable and maintainable, but does require more execution overhead than duplicated code.

An inline routine allows code for a routine to be duplicated in the place that it is called. It also allows parameters to be passed to such a routine.

In HP Pascal, an inline routine is declared with OPTION `inline` following the routine parameter list.

The use of inlined routines is a performance-for-space tradeoff. Consequently, these routines should be short and include only the code for the most frequently taken path. Large blocks of code that handle special cases should be made into routines that are called from the inlined routine.

The following are examples of inlined procedures for pushing and popping items from a stack.

<u>Example</u>

```
PROCEDURE push (item : itemtype)
   OPTION inline;
BEGIN  {push}
IF tos = topofstack THEN
   setuperror (stackoverflow)
ELSE
   BEGIN
   tos := tos + 1;
   stack[tos] := item;
   END;
END;  {push}

PROCEDURE pop
   OPTION inline;
BEGIN  {pop}
IF tos = bottomofstack
THEN
   setuperror (stackunderflow)
ELSE
   tos := tos - 1;
END;  {pop}
```

These examples show that the error conditions, stack overflow and stack underflow, are handled by calls to the procedure `setuperror`. The bodies of these procedures are very simple.

Other potential uses of inlined routines include performing operations such as exponentiation and exclusive-or which are not defined in Pascal.

Dynamic Routines

Procedure and Function Types

HP Pascal has been extended to to include procedure and function types. Procedure and function types are used to declare routines which are dynamically invoked at run-time. Procedure and function types are also called routine types.

Routine types are defined in the TYPE section. A routine type has no routine name associated with it. It only has its parameters, if any, in its parameter list.

A routine variable is a variable of a routine type. It is assigned a value by calling the predefined procedure `addr` on an actual routine. The actual routine must have parameters which are congruent to the parameters of the routine type. The rules for congruency are the same as those for procedural and functional parameters and are described in the *HP Pascal Reference Manual.*

In addition, for a function type, the type of the actual function return must be identical to that of the function type.

The predefined procedure `call` is used to invoke an actual procedure. The first parameter to `call` is a procedure variable or the result of the predefine `addr` on the actual procedure. The remaining parameters are the actual parameters corresponding to the parameters declared for the procedure type, if any.

Similarly, the predefined function `fcall` is used to invoke an actual function. The parameters to `fcall` are analogous to those of `call`.

The following is an example using these routine types.

<u>Example</u>

```
PROGRAM procedure_and_function_type;
TYPE
   ptype = PROCEDURE (i : integer);
   ftype = FUNCTION : integer;
VAR
   pvar : ptype;
   i    : integer;
PROCEDURE proc (i : integer); external;
FUNCTION func : integer; external;
BEGIN
pvar := addr (proc);
call (pvar,1);
i := fcall (addr (func));
END.
```

In this example, the type declaration `ptype` declares a procedure type with one value parameter of type `integer`. The type declaration `ftype` declares a function that returns an integer type. It has no parameters. The variable pvar is of type ptype.

Addr is called to create a reference to procedure proc and the value is assigned to the variable pvar. The procedure proc is invoked by the predefine call. The parameters to call are the procedure variable pvar and the value 1 for the integer value parameter of the procedure type ptype.

The function func is invoked by the predefine fcall. The parameter to fcall is the result of addr applied to the function func. There are no other parameters because the function type ftype has no parameters.

Unresolved Routines

HP Pascal allows a routine to remain unresolved through the link and load process.[2] At runtime, the predefine addr may be called to determine if an unresolved routine has been resolved. If the routine has been resolved, it may be invoked with the predefines call or fcall.

An unresolved routine is declared with OPTION unresolved following a routine parameter list. The EXTERNAL directive must also be used. It must be a level one routine.

The following is an example of invoking unresolved routines.

2Unresolved routines are not supported on HP-UX systems.

Example

```
$standard_level 'ext_modcal'$
PROGRAM option_unresolved;
VAR
  pvar1 : procedure;
  pvar2 : procedure;
PROCEDURE proc1
   OPTION unresolved;
   external;
PROCEDURE proc2
   OPTION unresolved;
   external;
BEGIN
pvar1 := addr (proc1);
pvar2 := addr (proc2);
IF pvar1 <> nil THEN
   call (pvar1)
ELSE IF pvar2 <> nil THEN
   call (pvar2);
END.
```

In this example there are two unresolved procedures, proc1 and proc2. Neither procedure has any parameters. The predefine addr is called to determine if these procedures are resolved. The check for nil is to verify that addr as returned a valid value.

Exception Handling

When a program is running four forms of exceptions may occur. These forms are: hardware errors, operating system errors, HP Pascal run-time errors, and programmer-defined errors.

In HP Pascal, a TRY-RECOVER block statement has been added to handle these exceptions. On an MPE/V system, the only way to trap runtime exceptions is to use intrinsics such as xlibtrap, xaritrap, and xsystrap.

The TRY-RECOVER construct consists of two parts: the TRY block and the RECOVER statement. In other words, for each TRY block there must be an associated RECOVER statement. A BEGIN END is not needed in the TRY block but is necessary for multiple statements in the RECOVER part.

When executing the statements in the TRY block, execution transfers to the RECOVER statement if an exception is raised. If no exception occurs in the TRY block, execution transfers to the statement following the RECOVER statement.

A user-defined exception is raised by calling the predefined procedure escape with a value for the exception. In the RECOVER statement, the value of the exception is accessed by calling the predefined function escapecode.

The following is an example of a programmer-defined exception:

Example

```
PROCEDURE try_recover (parm : integer);
CONST
   lessthan = 0;
   greaterthan = 1;
TYPE
   small = 0..10;
VAR
   local : small;
BEGIN
TRY
   IF parm < 0 THEN
      escape (lessthan)
   ELSE IF parm > 10 THEN
      escape (greaterthan)
   ELSE
      local := parm
RECOVER
   CASE escapecode OF
      lessthan : writeln ('< 0');
      greaterthan : writeln ('> 10');
      END;
writeln ('done');
END;
```

In this example, there are two exceptions. One exception is that `parm` is less than 0. The other exception is that `parm` is greater than 10. If either exception is encountered, the call to `escape` causes execution to transfer to the RECOVER statement. If the value of the escape code is `lessthan` the string '< 0' is written, if `greaterthan`, '> 10' is written.

If neither of the exceptions is encountered, the variable `local` is assigned the value of `parm` and execution transfers to the statement after the RECOVER part, namely the `writeln ('done')` statement.

This example can also take advantage of the Pascal run-time range checking if the programmer does not care whether the error was less than 0 or greater than 10.

Example

```
PROCEDURE try_recover (parm : integer);
TYPE
   small = 0..10;
VAR
   local : small;
   escapeval : integer;
BEGIN
TRY
   local := parm;
RECOVER
   BEGIN
   escapeval := escapecode;
   writeln (escapeval);
   END;
END;
```

In this example, the try block contains only the assignment statement. If parm is not within the range 0..10 an HP Pascal run-time exception is raised and the escape code is set. When execution transfers to the RECOVER part, the predefine escapecode accesses the HP Pascal escape code and the its value is written.

Note that in the RECOVER part, the value returned from the predefine escapecode is assigned to a local variable escapeval. This is a necessary precaution because system-level escapes may change the escape code. In this example, the call to writeln results in a system fwrite call which may modify the escape code.

On an MPE/XL system, the run-time escape codes for HP Pascal are available in the file PASESC.PUB.SYS.

The above examples are examples of local escapes. A local escape is an escape invoked within the static scope of the TRY block. In other words, it is an escape invoked within the statements in the TRY block.

Raising exceptions is not limited to local escapes. An escape may occur anywhere within the dynamic scope of the TRY block. That is, an escape may also occur within a routine called from a statement in a TRY block. This form of escape is called a nonlocal escape. Raising an exception in the dynamic scope of the TRY block also causes execution to transfer to the RECOVER statement. When more than one TRY block is active, execution transfers to the innermost RECOVER statement.

Example

```
PROCEDURE try_recover (parm : integer);
CONST
   lt0  = 0;
   gt10 = 1;
   gt5  = 2;
TYPE
   small = 0..10;
VAR
   local : small;
PROCEDURE inner_proc (parm : integer);
   BEGIN
   IF parm > 5 THEN
      escape (gt5);
   END;
BEGIN
TRY
   IF parm < 0 THEN
      escape (lt0)
   ELSE IF parm > 10 THEN
      escape (gt10)
   ELSE
      BEGIN
      inner_proc (parm);
      local := parm
      END
RECOVER
   CASE escapecode OF
      lt0  : writeln ('< 0');
      gt10 : writeln ('> 10');
      gt5  : writeln ('> 5');
      END;
writeln ('done');
END;
```

In the procedure `inner_proc` an exception is raised if `parm` is greater than 5. In this case the assignment of `parm` to `local` in the TRY block does not occur, and execution transfers to the recover statement which handles three exceptions. If `parm` is in the range 1..4, no exception occurs and execution continues at the assignment statement of `parm` to `local` and then jumps to the `writeln (done)` statement following the RECOVER statement.

Move Routines

There are three predefined procedures, `move_fast`, `move_l_to_r`, and `move_r_to_l`, for efficiently moving data from one array (source array) to another array (target array). The move predefines require that the element type of the source and target arrays be identical. Type coercion may be used to copy arrays that have different element types.

These predefines have five parameters:

```
move_fast   (n, source, soffset, target, toffset)
move_l_to_r (n, source, soffset, target, toffset)
move_r_to_l (n, source, soffset, target, toffset)
```

These parameters are:

```
n          Number of elements to move
source     Source array
soffset    Source offset
target     Target array
toffset    Target offset
```

The source and target arrays may be the same array. The differences in these predefines stem from the assumption regarding the addresses of the source and and target arrays. Move_fast assumes that the source array address does not overlap the target array address. In other words, the programmer is not depending on the rippling of data. Move_l_to_r and move_r_to_l do not have this assumption. Move_l_to_r performs a left to right component move from the source address to the target address. Move_r_to_l performs the move from right to left.

Example

```
$standard_level 'ext_modcal'$
PROGRAM move_routines;
TYPE
   rec = RECORD
     f1 : shortint;
     f2 : shortint;
     END;
   recarrtype = ARRAY [1..5] OF rec;
   intarrtype = ARRAY [1..5] OF integer;
VAR
   recarr : recarrtype;
   intarr : intarrtype;
BEGIN
intarr[1] := 0;
move_l_to_r (4, intarr, 1, intarr, 2);
$push, type_coercion 'representation'$
move_fast (5, intarr, 1, intarrtype(recarr), 1);
$pop$
END;
```

The move_l_to_r statement uses the rippling effect to initialize the elements of intarr to 0. The following move_fast uses intarr to initialize recarr. Type coercion is used to coerce recarr to intarrtype because the elements of recarr and intarr are different. Since the element type of recarr is a record, each shortint field of the record is initialized to 0.

Building Intrinsic Files

HP Pascal provides the facility for creating, modifying and listing an intrinsic file for HPPA systems. An intrinsic file is called a SYSINTR file on HPPA systems. On a MPE/V system, the program BUILDINT.PUB.SYS is available to add intrinsic declarations to an intrinsic (SPLINTR) file.

The compiler option `buildint` is used to build or modify an intrinsic file. The file to be built or modified is specified in the string associated with the buildint option. On MPE/XL the default intrinsic file is SYSINTR.PUB.SYS if no intrinsic file name is specified. Note that, in this case, the program must have write access to SYSINTR.PUB.SYS.

Each routine declared in a program with the buildint compiler option is added to the intrinsic file. Information about each declared routine and its parameters is added, as well. If a routine with the same name already exists in the intrinsic file, the new declaration replaces the one in the intrinsic file.

Routines are declared with the EXTERNAL directive. The parameter mechanisms for extensible and default parameters may be used. The language specification on the external directive may also be used.

A program with `buildint` is similar to any other HP Pascal program except that there are only external declarations and no main body.

Only certain Pascal types may be used as intrinsic parameter types. In general, the types that may be used are limited to those which are available in most languages supported by Hewlett-Packard. These types are described in the *HP Pascal Programmer's Guide*.

Example

```
$buildint 'sysintr'$
$standard_level 'ext_modcal'$
PROGRAM build_intrinsic_file;
TYPE
   pac = PACKED ARRAY [1..1024] OF char;

PROCEDURE xxx (VAR x1 : pac; x2 : integer);
   external;

PROCEDURE yyy (ANYVAR y1 : pac; y2 : integer)
   OPTION default_parms (y1 := nil, y2 := 0)
          uncheckable_anyvar;
   external;

PROCEDURE zzz (parm1 : integer; parm2 : integer);
   external ftn77;

BEGIN
END.
```

In this example, the intrinsic file name is `sysintr` in the user's group and account. Three procedures `xxx`, `yyy` and `zzz` are added to the intrinsic file. Procedure `xxx` is a simple declaration which does not use any new system programming features. Procedure `yyy`, in contrast, uses OPTION `default_parms` and OPTION `uncheckable_anyvar`. Procedure `zzz`, according to the language directive, is a FORTRAN77 subroutine.

The contents of a SYSINTR file may be listed with the compiler option `listintr`. If no string parameter is supplied to `listintr` the contents of the SYSINTR file is output to the formal designator `paslist`.

The information in a SYSINTR file may be accessed with the `INTRINSIC` directive. Each intrinsic declaration accesses the intrinsic file specified in the `sysintr` compiler option. If the compiler option `sysintr` is not specified, the default intrinsic file is SYSINTR.PUB.SYS.

Example

```
$sysintr 'sysintr'$
PROGRAM intrinsic_calls;

VAR
  a : PACKED ARRAY [1..1024] OF char;
  i, j : integer;

PROCEDURE xxx; intrinsic;
PROCEDURE yyy; intrinsic;
PROCEDURE zzz; intrinsic;

BEGIN
xxx (a, j);
yyy (i);
zzz (i, j);
END.
```

When the INTRINSIC directive is encountered, the HP Pascal compiler accesses the information in the intrinsic file 'sysintr' and uses it for checking actual parameters and for code generation.

For procedure yyy, i is a legal actual parameter for the first parameter because it is an anyvar parameter. The length of i is not passed, however, in the call to yyy, because it is an uncheckable_anyvar parameter. The default value 0 is passed for the second parameter since it was not supplied by the caller.

In the case of procedure zzz, the compiler knows that it is a FORTRAN77 subroutine and provides reference parameters even if they were declared as value parameters in the external declaration.

Conclusion

This paper has highlighted the new systems language features available in HP Pascal. You are encouraged to try these features when writing new HP Pascal programs or enhancing current programs.

References

HP Pascal Reference Manual (31502-60005)
HP Pascal Programmer's Guide (31502-60006)

Acknowledgments

I would like to thank Jon Henderson, Ron Rasmussen, Jean Danver, and members of the Pascal project, past and present, who reviewed this paper.

New Features of the MPE XL User Interface

by Thomas Shem & Jeff Vance
Hewlett-Packard Company
19447 Pruneridge Avenue
Cupertino CA, 95014

Introduction

The User Interface to the MPE operating system is the sum of the components which allow users to direct the HP 3000 in its execution of given tasks. The MPE V User Interface, consisting of commands, job control words (JCWs) and intrinsics, provides the user with the basic mechanisms to accomplish jobs. The User Interface to MPE V, while functional in nature, can hardly be considered "user friendly". Although some operations can be accomplished in a straightforward manner, the user must frequently rely on programs to perform tasks.

Much of this has changed with the MPE XL User Interface. This new User Interface while designed to be compatible with MPE V, was also designed to provide a powerful, flexible, and productive environment for the general user. The experienced user will find many new features which expose simple straightforward solutions to previously tedious and complex problems. Tasks, which on MPE V, required programs to solve, may now be accomplished though the new User Interface.

This paper describes the new MPE XL User Interface, focusing on the extensions beyond MPE V/E. The command language aspect of the User Interface is emphasized, and examples are provided to illustrate some of the simple straightforward solutions previously thought tedious and complex on MPE V.

The Command Interpreter

The Command Interpreter (CI) is central to the user interface. It is the mechanism whereby users of the HP 3000 access system functionality. An examination of the MPE V CI and the MPE XL CI will show how the user environment has been enhanced. It will also show how those enhancements create an environment which assists user.

The MPE V CI is a line oriented interface through which user commands are routed to the appropriate command executor (see figure 1). The status of the command string routing and subsequent command execution are indicated through JCWs (job control words).

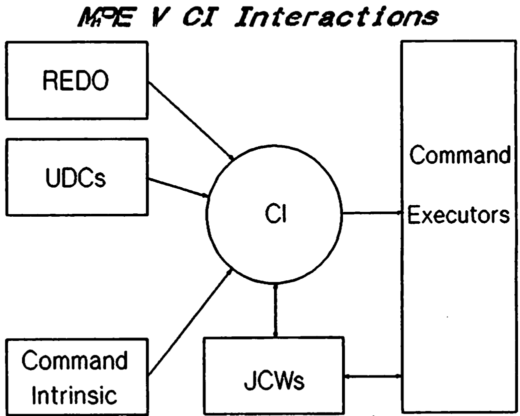


Figure 1. MPE V CI Interactions

User command strings may be fed to the CI through a variety of paths (see figure 1). Command strings can be input directly from a job or session, reissued or modified via the REDO command, invoked from a predefined UDC (user defined command), or issued from a user program or subsystems via the COMMAND intrinsic.

The MPE V Command Interpreter performs three major functions (see figure 2). Its first function is to interpret the input command string. The CI analyzes the command string for valid commands (either MPE commands or user defined commands). Another function of the CI is to determine if the user has the proper attributes to use the command. The CI compares the user capabilities against those assigned to the command. Lastly, the CI invokes the appropriate command executor. The executor completes the parsing of the command string, then performs the desired function.

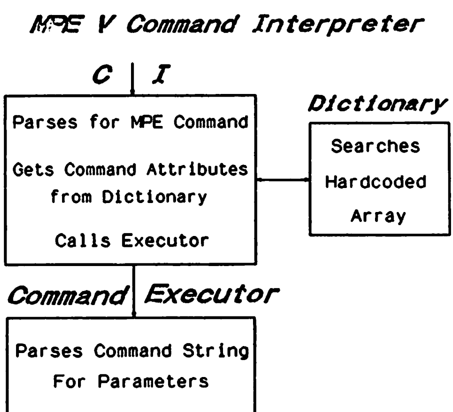


Figure 2. MPE V Command Interpreter

The MPE XL Command Interpreter performs the same basic functions as that of the MPE V Command Interpreter, since it was designed to be externally similar. The MPE XL CI has three major differences from the MPE V CI: it is implemented as an executable program instead of as a system process; the interfaces to the XL CI have been expanded and improved; and the structure of the CI has changed.

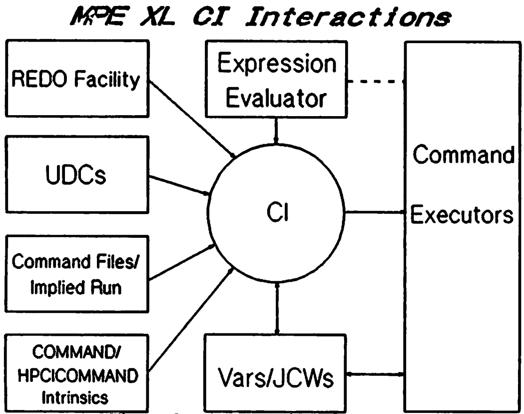


Figure 3. MPE XL CI Interactions

The MPE XL CI interacts with the same components as those on an MPE V system (see figure 3). However, these components have been specialized to facilitate usability. Command strings are still fed to the CI through a variety of paths. They can be input directly from a job or session, reissued or modified via the new REDO facility, invoked from predefined UDCs or command files (discussed later) or issued from the COMMAND or HPCICOMMAND intrinsics.

The XL CI has been restructured (see figure 4). It now consists of two parts; a centralized scanner/parser part and a command interpreter part. Command strings are first checked for correct syntax in the scanner/parser. The correct command executor is then invoked if the syntax is valid.

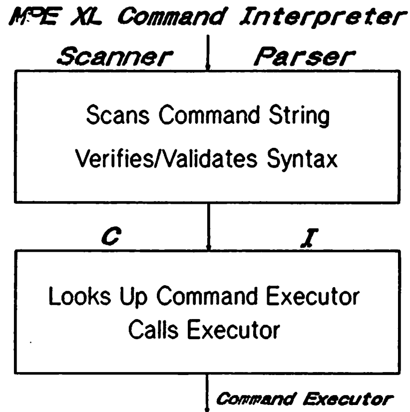


Figure 4. MPE XL Command Interpreter

The new structure of the MPE XL CI separates and specializes the parts of the system which deal with user interactions. This specialization gives the XL User Interface the ability to become a powerful tool which can be used when directing the system in a given task.

The User Environment

As can be seen from the two CI interaction diagrams (figures 1 and 3), many of the mechanisms existing in MPE V have been enhanced. These enhancements serve to make the XL CI more powerful, more flexible and much easier to use than its predecessor. A comparison of the differences in these features will show how they benefit the user, and serve to show how they interact to form a customizable user environment.

JCWs vs. Variables

Local to the user's sessions and jobs on MPE V are job control words. These words are temporary numeric indicators which provide statuses to the user whenever a command is executed (i.e. CIERROR). Additionally, users may use them to indicate the status of a program's execution. MPE V also provides several pre-assigned JCWs to indicate system information (HPMONTH, HPDAY, HPYEAR, etc.). JCWs are essential for controlling the flow of command execution within large batch jobs or complex User Defined Commands (UDCs).

In MPE XL, the idea of providing users with temporary variables has been expanded to include data types other than numeric. Variables can retain data in boolean form, 32-bit numeric form, string form, and JCW format.

These new variable types have also made it possible for MPE XL to provide more complete system information. The data in these predefined variables range from system global information (i.e. system time, system date, job count, and jobfence) to job or session system infomation (i.e. logon ID, capability lists, $STDIN ldev, $STDLIST ldev, interactive state, and CPU time used). Additionally, some variables are used to control the user environment. Modifying these variables changes aspects of the session environment. For example, when the variable HPAUTOCONT is set to TRUE, the effect is the same as if a CONTINUE statement preceded every command.

To support the variables, three new commands are provided: the SHOWVAR command displays variable and JCW information information; the SETVAR command assigns values to variables; and the DELETEVAR command removes variables. The SETJCW and SHOWJCW commands can be thought of as a subset of the variable commands.

The value of a variable is referenced by preceding the variable name with a "!" (e.g. !HPACCOUNT). The referenced variable is then replaced with its value (e.g. !HPACCOUNT is replaced by SYS if the user is logged into the SYS account). This is called dereferencing the variable (specifically, explicit dereferencing).

Variables may be dereferenced in two additional ways, implicit and recursive. Implicit dereferencing is simply the substitution of a variable's value for its name.

An example of implicit dereferencing occurs in the statement:

```
IF VAR = 0 THEN ...
```

In the above example the variable name (VAR) is replaced by its value without requiring a "!".

Recursive dereferencing of variables occurs when a variable contains the name of another variable. The following statement personalizes the CI's prompt to contain your username followed by "(current system time):", e.g "TOM (12:01):". The predefined variables HPUSER, HPHOUR, and HPMINUTE contain the username, the current system hour, and the current system minute.

```
:setvar hpprompt "!hpuser (!!hphour:!!hpminute):"
```

The expression to be assigned to the global variable is evaluated in the following manner:

- the quotes cause the variable's type to be a string

- !hpuser is explicitly dereferenced as the user ID

- the " (" is evaluated as a string literal

- the doubled exclamation points ("!!") have a special meaning and are used to represent a single exclamation point (i.e. they are folded into one; however, in general an even number of exclamation points do not cause dereferencing while an odd number of exclamation points do cause dereferencing - pairs are then folded) and are evaluated as a string containing a "!".

- the "hphour:" is evaluated as a string (combining with the previous exclamation point to form a dynamic variable; a variable within a variable)

- the doubled exclamation points are again folded into a single exclamation point, then evaluated as a string

- the "hpminute):" is evaluated as a string.

The string stored into HPPROMPT then looks like this:

```
TOM (!hphour:!hpminute):
```

When the variable HPPROMPT is referenced (it is referenced by the CI before displaying the prompt), recursive dereferencing occurs on the !hphour and !hpminute to obtain the current values stored in the system. As can be seen in the example, recursive dereferencing becomes very handy when the value of a dynamic variable is needed.

REDO vs The REDO Facility

One of the most useful commands available to the user of MPE V is the REDO command. Users do all sorts of amazing things with this command: re-execute their last command; correct errors in their last command with the editing features of the command; capture a previously typed command with the enter key, edit the captured command string then re-execute it; and capture some data (such as the capability list from the ALTUSER cierror message), add a valid command to the list then execute it. The main benefit of this command is that it saves the user from excess typing. The main drawback is that the user is limited to the last command entered.

On MPE XL, this command has grown into a facility of its own. The REDO facility now encompasses a command history stack, two new commands and enhanced editing features. The REDO buffer has been replaced by a command history stack. Whereas in MPE V only one command was able to be saved, the history stack allows users to save up to 1000 previous commands (although the default is 20). The history stack was designed to be user configurable, so users can control the depth of their own command history stack through the variable HPREDOSIZE. The history stack is supported by the new LISTREDO and DO commands and also the enhanced REDO command.

The LISTREDO command gives the user the option of defining the range of history stack to be displayed, and the option of referencing previous commands in one of three manners. Commands may be listed either numbered relative to the top command in the stack, or numbered relative to the first command entered during the session, or unnumbered.

For example:

```
:LISTREDO ;ABS             :LISTREDO ;REL             :LISTREDO ;UNN
     1) commandone              -3) commandone              commandone
     2) commandtwo              -2) commandtwo              commandtwo
     3) LISTREDO ;ABS           -1) LISTREDO ;REL           LISTREDO ;UNN
```

Re-execution of commands from the history stack is accomplished with the DO and REDO commands. The two commands are identical in nature, except that the REDO command allows interactive editing. Previous commands may be executed by using a relative or absolute command number.

Additionally, prior commands may be retrieved by dereferencing an absolute or relative history stack command number (e.g. !1 or !-2). The absolute command number must be within the range available on the current history stack. The predefined variables HPREDOSIZE and HPCMDNUM are provided to show the current redo stack size and current absolute command number.

Editing of previous commands may be accomplished by appending an edit string to the DO or REDO commands, or by using the interactive method, similar to that available on MPE V, with the REDO command. The basic editing directives available on MPE V are all available: i (insert), r (replace), d (delete), and u (undo) Several new directives have also been added: d> (delete to end of line), > (append to end of line), >d (delete from the end of line), >r (replace at the end of line), and c (change one string for another).

For example:

```
:DO ;EDIT=">dddd"
```

will result in the following, given any of the three previous :LISTREDO examples:

```
:LISTREDO                  :LISTREDO                  :LISTREDO
     1) commandone              1) commandone              1) commandone
     2) commandtwo              2) commandtwo              2) commandtwo
     3) LISTREDO ;ABS           3) LISTREDO ;REL           3) LISTREDO ;UNN
     4) LISTREDO                4) LISTREDO                4) LISTREDO
```

UDCs vs. Command Files

Whereas REDO is one of the most useful commands on MPE V, UDCs (User Defined Commands) are one of the most used feature. UDCs allow users to build a set of personalized commands since they are executed before MPE commands. They provide users with the ability to override or supersede MPE commands. UDCs offer a method for simplifying MPE commands. A sequence of commonly used commands can be bundled into one user command, simplifying user invocation and execution of tasks. Automatic invocation of a UDC at logon time can set up the user environment or be used to restrict users to a particular environment. Perhaps the most useful aspect of UDCs is that they provide a mechanism to avoid typing complex instructions.

With all of these useful functions, there wasn't much for MPE XL to improve upon. However, MPE XL has added several new features to UDC control and maintenance. Two new options have been added which control whether UDCs may recursively call themselves (OPTION RECURSION/ NORECURSION), and whether UDCs may be executed programmatically (OPTION PROGRAM/NOPROGRAM; more on this later). Maintenance of UDCs has been simplified via the new ;APPEND, and ;DELETE parameters to the SETCATALOG command. UDC files may be appended or deleted without having to uncatalog a session's current UDCs.

In MPE XL, an additional method for users to define their own set of customized commands has been provided through command files. Command files are simply files which contain one or more commands, much like UDCs. Commands within a command file may be MPE commands, UDCs, or filenames. They are similar to UDCs in all respects except for three differences:

- Command files are searched for after UDCs and MPE commands.

- Command files don't have the requirement of having to be cataloged.

- Command files do not support the control of recursion that UDCs provide via the RECURSION option, or the control for logon invocation via the LOGON option. In the case of recursion, the command file name need only be specified within the same command file in order to invoke recursion (UDCs require that the RECURSION option be specified). In the case of logon invocation, command files can not be invoked automatically at logon (except via a logon UDC).

Command files, like UDCs, may accept parameters by defining them in the header line and may use options. Invocation of command files, unlike UDCs, is executed by entering the filename.

Implied Run and Search Path

Program files like command files may now be invoked by entering the program filename. This is referred to as "implied run". The RUN command need no longer be specified to execute a program file in most cases. The optional INFO and PARM parameters are supported through the implied run. Use of any of the other optional parameters when invoking a program file will require that the RUN command be specified.

To enhance the user environment and to support invocation of command files and program files, a predefined variable HPPATH is provided to allow users to define a search path for the CI. The CI will follow the search path when a command has not matched a UDC or MPE command. The HPPATH variable virtually eliminates the need to explicitly specify a group or account with the command or program file name.

For example, suppose the HPPATH variable is set to "pub.sys,pub,!HPHGROUP, !HPGROUP" and a non-UDC/non-MPE command is entered. The CI will use the search path specified in the HPPATH variable to attempt to find a command or program file to execute. The CI will first look in the "pub.sys" group/account, then the "pub" group of the current logon account, then the user's home group, and finally the user's current logon group. Failing to find a command or program file at this point will result in an "UNKNOWN COMMAND NAME" message.

CI Flow Control Structures

The control of the sequence for execution of command statements in MPE V consists of the simple branching mechanism provided by the IF .. THEN .. ELSE .. ENDIF commands. As every programmer knows, branching mechanisms are adequate to get the job done, but just barely.

MPE XL expands upon this simple control structure by introducing a looping structure and a recursion structure. The WHILE .. DO .. ENDWHILE commands provide the users with a much needed control structure to repeat tasks. Another structure to accomplish looping as well as recursive executions is now supported through UDCs and command files. UDCs may invoke themselves if the RECURSION option is specified. Command files may also invoke recursion by using the filename within the command script. A predefined variable, HPUSERCMDEPTH, is provided to indicate the depth of nested UDCs and/or command files.

Additionally, a new RETURN command has beed provided to cause the execution of a User Command (UDC or command file) to return to the calling environment. User Commands invoked from the CI will return to the CI. Nested User Commands will return to calling UDC or command file.

Whereas in the past the command structure provided by MPE V did not allow users to get into complicated situations, the command structures provided by MPE XL can easily get the inexperienced user into a lot of trouble. Care should be used to prevent endless looping. Break will usually end an endless loop.

Expression Evaluator

In MPE V expressions are used in the IF .. THEN .. ELSE .. ENDIF control structure to indicate the truth or falsehood of the control structure. Expressions on MPE V consisted of the comparison of a JCW to another JCW, or a JCW to a fixed numeric value.

On MPE XL, evaluation of expressions has been separated into a new facility referred to as the expression evaluator, which provides the user with a rich set of features.

The expression evaluator supports a large set of functions. Many arithmetic operations are allowed, including: absolute value (ABS), modulo (MOD) and exponentiation (^). Many string operations are defined, such as: concatenation (+), length (LEN), ordinal (ORD), extraction (LFT, RHT, POS, STR), and case shifting (DWNS, UPS). Special variable functions include: existence (BOUND) and type (TYPEOF). Bit operations include: bitwise and (BAND), bitwise or (BOR), bitwise not (BNOT), bitwise exclusive or (BXOR), shift left (LSL), and shift right (LSR). Numeric conversion functions are: convert to octal (OCTAL) and convert to hexadecimal (HEX). Finally, some special functions provided via FINFO include: file existence (FINFO(0)), file creation date (FINFO(6)), file modification date (FINFO(8)), file code (FINFO(9)), foptions (FINFO(13)) and others.

Expressions may be implicit or explicit. Implicit expressions are only available in in four commands: CALC, SETVAR, IF, and WHILE. Explicit expressions may be dereferenced in any command by enclosing the expression within square brackets and preceding it with an exclamation mark (e.g. ![expression], referred to as "expression substitution"). For example, !["li"+"stf"] will cause the LISTF command to be executed.

Mixed expressions are not allowed, for example, the expression "a" + 1 would result in an error. Standard precedence rules apply, with explicit variable dereferencing superseding all other operations.

Miscellaneous New Commands

Several new commands have also been provided to enhance the new User Interface. These include the CALC, CHGROUP, COPY, ECHO, INPUT, PRINT, and XEQ commands. A complete description of each of these commands can be found in the MPE XL Commands Reference Manual (Part Number 32650-90003); however, specific usage of some of these commands is described below:

- the CALC command provides easy access to the Expression Evaluator for quick evaluation of expressions.

- the CHGROUP command gives users an easy way to switch from one group to another, without logging off.

- the COPY command provides for simple fast file copies even in break mode.

- the ECHO command displays text to $STDLIST (this is particularly useful in conjunction with expression substitution, "![ ]", and explicit variable dereferencing).

- the INPUT command can be used to prompt a user for data, and is a convienent way to load a string variable. Also, note the timed read feature.

- the PRINT command provides a quick way to display a file.

- the XEQ command will execute a command or program file, even if the file contains the same name as a UDC or MPE command.

COMMAND vs HPCICOMMAND

Programmatic invocation of MPE commands on MPE V is accomplished via the COMMAND intrinsic. Unfortunately, the COMMAND intrinsic does not perform all the functions available through the MPE V CI. On MPE XL, the introduction of the new HPCICOMMAND intrinsic provides users with a programmatic method for accessing most of the MPE XL CI functions, as well as command files, UDC invocation, and implied run.

The Command Language

The combination of commands, variables, expression evaluations, control structures, UDCs, and command files all come together to form a command language. With MPE V, end-users needed system programmers to build complex programs to assist in the operation of their HP3000. Now, non-programmer end-users, as well as experienced users and programmers, will be able to control their system without writing a program. The "language" of MPE XL exposes the power of the HP3000 at the level of the command interpreter instead of hiding it.

To illustrate the power available through the MPE XL command language, several examples are provided. These examples show how common operations which are tedious or cumbersome can be made simpler.

1. "CENTER" - this command file is an example of some of the string manipulation functions supported by the expression evaluator. It will echo to $STDLIST a centered string. It is used later in the "CALCIT" example.

```
Parm string
COMMENT
COMMENT +-----------------------------------------------+
COMMENT | CENTER                                        |
COMMENT | Note: cent_spc is loaded with a blank string  |
COMMENT +-----------------------------------------------+
COMMENT
setvar cent_spc "                                              "
echo ![lft(cent_spc,(80-len("!string"))/2)]!string
deletevar cent_spc
comment end of center

Usage:
:center 'Center this string'
                              Center this string
:
```

2. "FKEY" - this command file is an example of how escape functions to a terminal can be done. It will set up one function key.

```
PARM KEY,L1=" ",S1="UD",A1=2
COMMENT
COMMENT +-----------------------------------------------+
COMMENT | FKEY                                          |
COMMENT | Sets A SINGLE function key in one invocation. |
COMMENT | The "L" parameter is the label, the "S" parameter |
COMMENT | is the string to be generated when the function |
COMMENT | key is pressed, and the "A" parameter is the  |
COMMENT | key attribute parameter, where 0=Normal, 1=Local, |
```

```
COMMENT | and 2=Transmit attributes for each key. The "KEY" |
COMMENT | is, of course, the key to set.                      |
COMMENT +----------------------------------------------------+
COMMENT
IF (HPJOBTYPE="S") AND (HPDUPLICATIVE=TRUE) THEN
  setvar esc chr(27)
    SETVAR vars "!S1"
    SETVAR varl "!L1"
    SETVAR lens LEN(vars)
    SETVAR lenl LEN(varl)
    SETVAR wkey "!esc"+"&f!a1"+"a!key"+"k"+&
      "!lenl"+"d"+"!lens"+&
      "L"+"!varl"+"!vars"
  echo !wkey
  ECHO !esc&jB
  DELETEVAR esc,vars,varl,lens,lenl,wkey
ENDIF
COMMENT end of fkey

Usage:
:fkey 1,test,'echo this is a test'
         now using the f1 key will perform the following:
:echo this is a test
this is a test
```

3. "TRIM" - this command file will trim all characters of a specified type from a variable name. It is an example of how string manipulation can be done.

```
parm varname,trimchar=" ",from=RIGHT
COMMENT
COMMENT +----------------------------------------------------+
COMMENT | TRIM                                               |
COMMENT | Trims all trimchar from varname, starting at from.|
COMMENT +----------------------------------------------------+
COMMENT
if not (bound(!varname)) then
   echo (TRIM): The variable !varname is not defined.
else
   if not(ups(lft('!from',1)) = 'L') then
      setvar trim_off 'RHT'
      setvar trim_save 'LFT'
   else
      setvar trim_off 'LFT'
      setvar trim_save 'RHT'
   endif
   while (len(!varname)>0) and (!trim_off(!varname,1)='!trimchar')
      setvar !varname !trim_save(!varname,len(!varname)-1)
      endwhile
   deletevar trim_off,trim_save
endif
comment end of trim
```

```
Usage:
:setvar string 'this string needs the question mark stripped off?????'
:trim string,?
:showvar string
STRING = this string needs the question mark stripped off
```

4. "ADDCAP" - this command file provides a simple method for adding capabilities to a "user"s existing capability list. Note the re-logging on option at the end of the command file.

```
parm cap=''
COMMENT
COMMENT +-----------------------------------------------------------+
COMMENT |ADDCAP                                                     |
COMMENT |Adds a new capability to the "user"s existing capabilities. AM |
COMMENT |is required to execute the :ALTUSER command. The new capability|
COMMENT |is only available after re-logging on, which the user will be  |
COMMENT |prompted for.                                              |
COMMENT +-----------------------------------------------------------+
COMMENT
if ('!cap' = '') then
   echo (ADDCAP): Your capabilities are: !hpusercapf.
   return
endif
if (pos('![ups('!cap')]','!hpusercapf') <> 0) then
   echo (ADDCAP): You already have     : !cap.
   echo (ADDCAP): The capabilities are : !hpusercapf.
   return
endif
setvar addcap_temp "!hpusercapf,!cap"
setvar cierror 0
continue
altuser !hpuser;cap=!addcap_temp
if cierror <> 0 then
   echo (ADDCAP): The capabilities remain: !hpusercapf.
else
   setvar addcap_temp,ups(addcap_temp)
   echo (ADDCAP): !hpuser new capabilities are: !addcap_temp.
   setvar addcap_temp 'N'
   input addcap_temp,'(ADDCAP): Log off/on now (Y/N) ==>',10
   if cierror = -9003 then
      comment: Timed read expired.
      echo
      echo (ADDCAP): Timed 10-second read expired. Logon cancelled.
   else
      if not(lft(ups(addcap_temp),1) = 'Y') then
         echo (ADDCAP): New capabilities take effect at next logon.
      else
         hello !hpjobname,!hpuser.!hpaccount,!hpgroup
      endif
   endif
endif
```

```
deletevar addcap_temp
COMMENT end of addcap

Usage:
:listuser foo
********************
USER: FOO.UI

HOMEGROUP:                    PASSWORD: **
MAXPRI    : 150               LOC ATTR: $00000000
LOGON CNT: 0
CAP: AM,BA,IA

:addcap
(ADDCAP): Your capabilities are: AM,BA,IA

:addcap am
(ADDCAP): You already have     : AM
(ADDCAP): The capabilities are : AM,BA,IA

:addcap ds
(ADDCAP): FOO new capabilities are: AM,BA,IA,DS
(ADDCAP): Log off/on now (Y/N) ==>n
(ADDCAP): New capabilities take effect at next logon.
```

5. "FINFO" - this command file will display the file label information for a given file. Note the use of finfo in the IF..THEN control structure.

```
PARM file
COMMENT
COMMENT +------------------------------------+
COMMENT | FINFO                              |
COMMENT | Use finfo to show file label info. |
COMMENT +------------------------------------+
COMMENT
if not(finfo('!file',0)) then
   comment  File does not exist.
   if lft('!file',1) <> '*' and lft('!file',1) <> '$' then
      comment  Qualify file before reporting non-existence.
      if pos('.','!file') > 0 then
         if pos('.',rht('!file',len('!file')-pos('.','!file'))) > 0 then
            echo ![ups('!file')] does not exist.
         else
            echo ![ups('!file')].!hpaccount does not exist.
         endif
      else
         echo ![ups('!file')].!hpgroup.!hpaccount does not exist.
      endif
   else
      echo !file does not exist.
   endif
else
```

```
   comment ** formal file designator **
   echo (FINFO): Full file description for ![finfo('!file',1)] follows:
   comment ** creator and create/modify dates **
   echo    Created by ![finfo('!file',4)] on ![finfo('!file',6)].
   echo    Modified on ![finfo('!file',8)] at ![finfo('!file',24)].
   comment ** file code **
   if finfo('!file',9) = '' then
      echo    Fcode: ![finfo('!file',-9)].
   else
      echo    Fcode: ![finfo('!file',9)] (![finfo('!file',-9)]).
   endif
   comment ** rec size, eof, flimit **
   echo    Recsize: ![finfo('!file',14)], Eof: ![finfo('!file',19)], &
Flimit:![finfo('!file',12)].
   comment ** foptions **
   setvar _fopt finfo('!file',-13)
   echo    Foptions: ![finfo('!file',13)] (#!_fopt, ![octal(_fopt)],&
![hex(_fopt)]).
   deletevar _fopt
endif
COMMENT  End of finfo.
```

```
Usage:
:finfo a
(FINFO): Full description for A.GROUP.ACCT follows:
   Created by TOM on SUN, MAY 29, 1988.
   Modified on SUN, MAY 29, 1988 at 3:21 PM.
   Fcode: 0.
   Recsize: -80, Eof: 5, Flimit:5.
   Foptions: ASCII, FIXED, NOCCTL, STD (#5, %5,$5).
```

6. "PURGESP" - this command file will purge multiple spool files. Note the multiple entry points and the use of the Command Interpreter, CI.PUB.SYS.

```
parm user="@",grp="pub",entry_point="purgesp",command=""

COMMENT
COMMENT +-----------------------------------------------------+
COMMENT | PURGESP                                             |
COMMENT | This file will purge all spool files which belong to the |
COMMENT | specified user (ie - @, @.@, mgr.test, etc.).       |
COMMENT | Only spool files in the READY state are purged.     |
COMMENT +-----------------------------------------------------+
COMMENT

COMMENT << this entry point is the main body >>
if "!entry_point" = "purgesp" then
  COMMENT <<cleanup temporary files from a previous run of this file>>
   if FINFO ("spin.!grp", 0) then
      purge spin.!grp
   endif
```

```
if FINFO ("spin2.!grp", 0) then
   purge spin2.!grp,temp
endif

if FINFO ("spout.!grp", 0) then
   purge spout.!grp
endif

if FINFO ("purgein.!grp", 0) then
   purge purgein.!grp
endif

if FINFO ("purgein2.!grp", 0) then
   purge purgein2.!grp,temp
endif

COMMENT << build temporary files >>
build spin.!grp;rec=40,,f,ascii;disc=10000
build spout.!grp;rec=40,,f,ascii;disc=10000
build purgein.!grp;rec=40,,f,ascii;disc=10000
file spin2.!grp;rec=40,,f,ascii;disc=10000
file purgein2.!grp;rec=40,,f,ascii;disc=10000

COMMENT << generate the input file for spook >>
run ci.pub.sys;info="purgesp ,,'getspin','s !user'" &
              ;stdlist=spin.!grp;parm=3

COMMENT << extract only the lines that contain spook commands >>
print spin.!grp,*spin2.!grp;start=3

COMMENT << get a list of the spool files to purge >>
run spook.pub.sys;stdin=spin2.!grp &
                 ;stdlist=spout.!grp

COMMENT << convert the list of files to the format "p o####" >>
run ci.pub.sys;info="purgesp ,,getxddno" &
              ;stdin=spout.!grp &
              ;stdlist=purgein.!grp &
              ;parm=3

COMMENT << extract only the lines that contain xdd numbers >>
print purgein.!grp,*purgein2.!grp;start=3

COMMENT << purge the selected spool files >>
run spook.pub.sys;stdin=purgein2.!grp

COMMENT << clean up >>
purge spin.!grp
purge spin2.!grp,temp
purge spout.!grp
purge purgein.!grp
purge purgein2.!grp,temp
reset spin2.!grp
```

```
   reset purgein2.!grp
else

  COMMENT << this entry point creates the file 'spin' >>
  if "!entry_point" = "getspin" then
     echo !command
     echo exit
  else

    COMMENT << this entry point converts a spook-generated list of >>
    COMMENT << spool files to a list of just xdd numbers preceded  >>
    COMMENT << by a 'p' to tell spook to purge the file            >>
    if "!entry_point" = "getxddno" then
       setvar hpmsgfence 2
       setvar eof false
       setjcw cierror 0

       while not eof
          continue
          input line

          if cierror <> 900 then
             if LFT (line, 2) = "#O" then
                if POS ('READY', line) > 0 then
                   echo p ![STR (line, 3, 6)]
                endif
             endif
          else
             setvar eof true
          endif
       endwhile

       echo exit
       deletevar eof
       deletevar line
       setvar hpmsgfence 0
    endif
  endif
endif
COMMENT end of purgesp
```

```
Usage:
:spook5
SPOOKHPE A.02.50 (C) HEWLETT-PACKARD CO., 1985
>s
#FILE   #JOB    FNAME    STATE  OWNER
#O1777  #J758   $STDLIST READY  TOM.UI
#O1778  #J759   $STDLIST READY  TOM.UI
#O1779  #J760   $STDLIST READY  TOM.UI
#O1780  #J761   $STDLIST READY  TOM.UI
#O1781  #J762   $STDLIST READY  TOM.UI
#O1782  #J763   $STDLIST READY  TOM.UI
>e
```

```
END OF PROGRAM
:purgesp tom


END OF PROGRAM


END OF PROGRAM


END OF PROGRAM

SPOOK A.10.10  (C) HEWLETT-PACKARD CO., 1983
> #FILE    #JOB    DEV/CL    SECTORS     OWNER
#01777  #758    LP           36      TOM.UI
> #FILE    #JOB    DEV/CL    SECTORS     OWNER
#01778  #759    LP           36      TOM.UI
> #FILE    #JOB    DEV/CL    SECTORS     OWNER
#01779  #760    LP           36      TOM.UI
> #FILE    #JOB    DEV/CL    SECTORS     OWNER
#01780  #761    LP           36      TOM.UI
> #FILE    #JOB    DEV/CL    SECTORS     OWNER
#01781  #762    LP           36      TOM.UI
> #FILE    #JOB    DEV/CL    SECTORS     OWNER
#01782  #763    LP           36      TOM.UI
>
END OF PROGRAM
:
```

7. "CALCIT" - this command file will allow a user to interactively use the expression evaluator. Note the high usage of screen control escape characters.

```
PARM enh_ch=D
COMMENT
COMMENT +-----------------------------------------------+
COMMENT | CALCIT                                        |
COMMENT | Interactive calculator using :calc and :input. |
COMMENT +-----------------------------------------------+
COMMENT
echo ![chr(27)+'h'+chr(27)+'J']
center 'MPE XL Interactive Calculator'
center ': executes any MPE command!!!'
center 'Type EXIT or [RETURN] to exit'
echo
setvar calcit_esc chr(27)+'C'
setvar calcit_expr 1
while calcit_expr < 7
  setvar calcit_esc calcit_esc+calcit_esc
  setvar calcit_expr calcit_expr + 1
endwhile
setvar calcit_esc chr(27)+'A'+calcit_esc
setvar calcit_prompt lft(ups("!-1"),pos(' ',"!-1"+' ')-1)+' ==> '
```

```
setvar calcit_expr 'Hello'
while (not ups(calcit_expr) = 'EXIT') and (not calcit_expr = '')
  setvar calcit_expr ''
  input calcit_expr,"!calcit_prompt"
  setvar calcit_len len(calcit_expr)
  while (len(calcit_expr) > 0) AND (lft(calcit_expr,1) = chr(32))
    setvar calcit_expr rht(calcit_expr,len(calcit_expr) - 1)
  endwhile
  if (not ups(calcit_expr) = 'EXIT') and (not calcit_expr = '') then
  if lft(calcit_expr,1) = ':' then
    continue
    ![rht(calcit_expr,len(calcit_expr)-1)]
  else
    setvar hpmsgfence 2
    setvar cierror 0
    continue
    setvar hpresult !calcit_expr
    if not cierror = 0 then
      setvar hpmsgfence 0
      continue
     calc !calcit_expr
    else
     echo ![lft(calcit_esc,2+(len(calcit_prompt)+calcit_len)*2)+&
           ' = '+chr(27)+'&d!enh_ch']!hpresult
    endif
  setvar hpmsgfence 0
  endif
  endif
endwhile
deletevar calcit_@
comment end of calcit
```

Usage:

```
:calcit
                      MPE XL Interactive Calculator
                      : executes any MPE Command!!!
                      Type EXIT or [RETURN] to exit

CALCIT ==> :listf a

FILENAME

A

CALCIT ==> 5+7
CALCIT ==> 5+7 = 12
CALCIT ==> exit
:
```

TRANSACT/XL: Strategy for Migration to Native Mode

by Ga.y Peck
Hewlett-Packard
19111 Pruneridge (Mail Stop 44MH)
Cupertino, CA 95014

* * * * *

ABSTRACT

Native Mode TRANSACT is a way to take full advantage of the capabilities of the Series 900. It compiles TRANSACT source or P-code into Native Mode object code.

This presentation covers:

Steps in the TRANSACT/XL migration process

Ease of conversion from MPE/V to MPE/XL and from CM to NM

Minor exception conditions that may require code modifications

Practical experience from actual application migrations

Expected performance

* * * * *

TRANSACT/XL COMPILER

In the past, a migration to a new computer has been a scary prospect. The question boils down to this: Is it possible to take thousands of lines of source code that were running on your existing computer, recompile it with a NEW compiler on a NEW computer, and have them run better than before with high reliability and exceptional performance? This is clearly possible with Hewlett-Packard's new TRANSACT/XL compiler. In addition, a carefully executed migration plan practically guarantees the success of the migration.

On the Series 900 machines, TRANSACT programs can run in Compatibility Mode (CM) or Native Mode (NM). Compatibility Mode is a program environment that executes classic 3000 instructions on the Series 900 machine. Native Mode uses the innate features of the new instruction set and the MPE/XL operating system to take full advantage of Precision Architecture. In CM, TRANSACT source files are processed by TRANCOMP into P-code files, which are then executed by the TRANSACT processor just as on classic 3000 machines under MPE/V. In NM, TRANSACT provides a compiler that reads either ASCII source files or TRANSACT P-code files and generates Native Mode object modules.

MIGRATION PROCESS

A successful migration will result by following the stages in the migration process: training, planning, preparation, installation, CM operation, and NM operation. Briefly, the migration site receives training on MPE/XL the migration process; the project goals, milestones, and tasks are planned; the staff completes tasks in preparation for arrival of the Series 900 equipment and software; the hardware, MPE/XL and subsystems are installed; the staff restores MPE/V applications and tests in CM; and the staff compiles and tests applications in NM. The migration process is covered in the tutorial presentation, "Steps to a Successful Migration" and in the XL migration guides.

I will focus on aspects of migration peculiar to TRANSACT. In the planning and preparation stages, migration tool utilities are usually used to flag potential incompatibilities with the NM side and to predict resource utilization. These utilities are not relevant to TRANSACT applications since TRANSACT applications are NOT made up of program files. Instead TRANSACT applications are groupings of P-code (IP) files (more like data) which are interpreted by the TRANSACT run-time processor. The only program files are TRANCOMP and TRANSACT themselves which have been converted and tested in CM for you by Hewlett-Packard. The TRANSACT/XL compiler runs in Native Mode.

There are few, if any, changes to be made when converting from a TRANSACT/V source program to a TRANSACT/XL source program. Therefore, NO automatic conversion utility has been provided. Changes are made manually with a text editor of choice. Virtually all TRANSACT features are supported by TRANSACT/XL. The few features not supported are:

Run-time resolution of data item definitions from a dictionary
Test modes
INITIALIZE built-in command
CALLs to TRANSACT/V, REPORT/V, or INFORM/V (TRANSACT/XL calls are supported)

TRANSACT language features that are specific to the MPE/V environment but not applicable in the MPE/XL environment are ignored by the TRANSACT/XL compiler (e.g., NOLOAD, SWAP). Some additional considerations for the PROC verb will be covered in a moment.

New features designed to maximize effectiveness of XL applications can be quickly incorporated. These are principally new compile options available in the "INFO=" string of the TRANSACT/XL compile commands, TRANXL, TRANXLLK, and TRANXLGO. These are the same kinds of compile commands used by other NM languages. The compile options, including OPTIMIZE and SUBPROGRAM, will be discussed later.

The existing TRANSACT language feature set, Dictionary/V, System Dictionary, and Native Language Support are supported at compile time. TurboIMAGE, MPE/V and MPE/XL file systems (KSAM and MPE files), VPlus, NLS, and both the IEEE and HP standard floating point formats are supported at run-time.

STEPS TO MIGRATE TRANSACT/V to TRANSACT/XL

1) RESTORE your TRANSACT/V source files onto the MPE/XL system.

2) TEST in Compatibility mode (CM). Look for "setup" mistakes because you are trying to duplicate your production environment on a new machine. Look for differences, if any, between the XL environment and the classic 3000.

   NOW on to Native Mode...

3) Examine each program for the following special conditions.

3a) Does it use the PROC verb to call system intrinsics?

   Make sure each intrinsic is defined using the DEFINE(INTRINSIC) statement or use the new compiler option PROCINTRINSIC. These measures are not required for TurboIMAGE and VPlus intrinsics.

3b) Does it use the PROC verb to call option-variable system intrinsics with 32-bit parameters? (i.e., FCHECK, FDELETE, FGETINFO, FOPEN, MYCOMMAND, STACKDUMP, WHO)

   Explicitly pass the 32-bit parameter. For example, in the following code, pass the "filesize" parameter replacing the two commas currently used to denote a null filesize with the filesize parameter and a single comma.

```
system exam1;
define(item) file-name x(20):
             foption   i(4):
             aoption   i(4):
             filenum   i(4):
             filesize  i(9):    <<32 bit integer>>
             bitmap    i(4);
define(intrinsic) fopen;
list file-name:
     foption:
     aoption:
     filenum:
     filesize,init:
     bitmap;
move (file-name) = "OLDFILE";
let (foption) = 5;               <<old ascii file>>
let (aoption) = 0;               <<read access>>
let (bitmap) = 7176;             <<111000001000 passing 1st three>>
                                 << and filesize; this bitmap is  >>
                                 << valid in CM, but is ignored in NM >>
proc fopen(%(file-name),#(foption),#(aoption),
        <<,,,,,,,,,,,, >>        <<old place-holding commas       >>
          ,,,,,,#(filesize),,,, <<each comma denotes a parameter >>
                                 <<note that there is one less comma>>
           &(filenum),#(bitmap));
```

3c) Does it use the PROC verb to call subroutines written in other languages?

The alternatives are to write a STUB for the routine or rewrite it in a Native Mode language. There may be differences between MPE/V based compilers and MPE/XL based compilers. Please refer to the individual language migration guides. For example, the MPE/XL based COBOL compiler converts hyphens to underscores. The MPE/V based COBOL compiler leaves hyphens as is.

3d) Does it use the CALL verb to call INFORM/V or REPORT/V?

Since this feature is not supported, you can choose from these workarounds (examples follow):

Continue to run the program in compatibility mode
Rewrite the INFORM/V report in Transact and compile it with Transact/XL
Use process handling to invoke Inform/V
Convert the report to BRW/XL and use the PROC verb to call the
BRW/XL intrinsics

The following is a TRANSACT program that executes a BRW report. Prior to running the program, a BRW report was designed and compiled into a BRW/XL execution file named BRWEXECR.

```
system brw1,vpls=myff(mainmenu(selection));
define(item) brw-comarea     x(106):
             brw-status      i(4)=brw-comarea:
             brw-error       i(4)=brw-comarea(3):
             brw-com-length  i(4)=brw-comarea(5):
             brw-exec-file   x(36)=brw-comarea(7):
             brw-defaults    i(4),init=0:
             selection       i+(1);
list brw-comarea:brw-defaults:selection;
let (brw-com-length) = 50;
get(form) mainmenu;
if (selection) = 1 then
   do
   proc brwinitrequest ((brw-comarea));
   move (brw-exec-file) = "BRWEXECR  ";
   proc brwstartrequest ((brw-comarea),(brw-defaults));
   display brw-status:brw-error;
   proc brwstoprequest ((brw-comarea));
   doend;
end;
```

3e) Does it use the CALL verb to call a TRANSACT/V program?

Compile the called TRANSACT/V program with the TRANSACT/XL compiler using the SUBPROGRAM option. Place the called program in an RL or XL to be resolved during linking or loading.

3f) Does it access files that contain real numbers?

The code generated by the TRANSACT/XL compiler supports both IEEE and HP floating point formats. Under NM MPE/XL, internal storage of real numbers is in the IEEE format. Translation between IEEE and HP formats from and to files and databases is done after the read and before the write on I/O. If no format is specified for a file or database, IEEE numbers are assumed. The compiler option HP3000_16 is available for defining a floating point format for all the files. If the floating point format for individual files is different from that specified by the compiler option, you can express the requirements in the FILE or BASE specification of the SYSTEM statement. This is done by putting HP3000_16/32 in the "file-option-list" or in the "basetype" (follows "optlock"). Because internal representation of real numbers is different between MPE/V and MPE/XL, individual values may change slightly during conversion.

The following program illustrates converting real numbers from the MPE/V format to the MPE/XL standard format. Note that the HP3000_16 option is applied to the input file and the HP3000_32 option is applied to the output file. This causes item-name "R4" to be read as an MPE/V format real number and to be written as an MPE/XL standard format real number.

```
system convrt,file=in(read(hp3000_16))
             ,file=out(write(hp3000_32));
define(item) x2 x(2):
             i4 i(4):
             i9 i(9):
             r4 r(4);
list x2:i4:i9:r4;
find(serial) in,perform=100-convert;
exit;
100-convert:
    put out;
    return;
```

3g) Does it resolve variable definitions at run-time?
(e.g., DEFINE(ITEM) itemname *;)

Define all variables at compile time.

3h) Does it rely upon the INITIALIZE command to execute the next program?

Change user procedures to exit the program and RUN a second program at the MPE/XL command level.

4) Determine which compile options are needed. Supply these in the "INFO=" parameter on the TRANSACT/XL compile commands.

Like compatibility mode TRANCOMP, the TRANSACT/XL compiler allows you to control certain compilation features by supplying compiler options via the INFO= parameter. These options can be included on any of the commands that are used to invoke the TRANSACT/XL compiler: TRANXL, TRANXLLK, TRANXLGO, and RUN TRAN.PUB.SYS. The new compiler options are:

DYNAMIC_CALLS generates dynamic calls for all CALL statements in the program. This allows a program to be executed even if some of the programs that it calls are not available at load-time.

HP3000_16 causes the program to use the HP floating point format for all files and databases. If the NOHP3000_16 option is specified, then all files are expected to use the IEEE floating point format.

PROCALIGNED_16, PROCALIGNED_32, PROCALIGNED_64 cause the compiler to assume that all 16/32/64-bit aligned parameters are correctly aligned on 16/32/64-bit boundaries. Using this option improves run-time efficiency, since the compiler only generates a run-time check to ensure that these parameters are correctly aligned.

PROCINTRINSIC option is identical in effect to declaring intrinsics with a DEFINE(INTRINSIC) statement, but is less efficient.

SUBPROGRAM is used when compiling a program to be called by another TRANSACT/XL compiled program. No outer block is generated. The TRANSACT/XL compiler creates a single RSOM file regardless of how many SYSTEM statements are in a source file. When a source file contains more than one system, the default is to compile the first SYSTEM encountered with option NOSUBPROGRAM and the remaining with the option SUBPROGRAM as they are assumed to be subprograms called by the first system. Using the SUBPROGRAM compiler option causes all the systems in the file to be compiled with the SUBPROGRAM option.

OPTIMIZE directs the compiler to generate level 1 optimized code. Using this option causes the compile to be slower, but produces object modules that are more efficient at run-time.

5) Compile the programs under TRANSACT/XL.

6) Examine the compile listings for errors.

7) Test applications as extensively as possible. If discrepancies or defects are identified, please verify these under the TRANSACT/V processor. Please do as much as you can to isolate new defects in your applications from those in TRANSACT/XL. Report defects to the Response Centers.

Our initial migration sites have been converting thousands of lines of code with few unexpected errors in the applications and almost no defects in TRANSACT/XL. Most snags that I have observed tend to fall into three categories: setup (wrong capabilities; UDC not set); XL learning curve (LINKEDIT; XL user libraries); and test suites not fully debugged on the classic 3000.

Because of the additional testing prescribed by the migration process, a new error is just as likely to be an undiscovered defect in the application as it is a defect in TRANSACT/XL. I emphasize the importance of using a fully tested test suite so that you are fairly certain of testing the application and not the test data. Some defects may be caused in the MPE realm and come to the surface in a TRANSACT program. Be aware of differences in subsystems as well. VPlus, TurboIMAGE, and network products all have their own migration guidelines. For example, in VPlus, performing character mode I/O while VPlus "owns" the terminal will cause VPlus or the driver to hang (Vturnoff and Vturnon intrinsics alleviate this).

Preliminary performance data shows that a TRANSACT program compiled in NM is performing within the same tolerances as other NM languages. A TRANSACT/XL performance test was just completed by a major HP customer. The application was an interactive materials inventory and maintenance system comprising 500,000 lines of TRANSACT code, 260 VPlus screens, and 6 TurboIMAGE databases. A menu was used to select functions performed by individual TRANSACT programs. The menu was compiled into a Native Mode main program and the TRANSACT subprograms were compiled into a XL library. The Series 950's maximum throughput at the saturation point was as much as 2.8 times the Series 70 running FASTRAN compiled TRANSACT code. The customer also found that the maximum number of recommended users supported by the S/950 was 2 to 3 times the practical limit of users on their S/70.

TRANSACT migration to the Series 900 should be smooth and straightforward. Customer enthusiasm is running high because of the exciting performance, product reliability, ease of migration, and expert support channels.

BIBLIOGRAPHY

TRANSACT/XL Migration Guide
TRANSACT/3000 Reference Manual, Rev. 10/87, Appendix H (future update)

TITLE: Effective Use of HP-PA Optimizing Compilers

AUTHOR: Brad Ahlf

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 2072

Programming for MPE XL Performance

Dave Trout
Hewlett-Packard Company
2 Choke Cherry Road
Rockville, MD 20850 USA

Introduction

Application performance issues are fairly well understood these days for the existing MPE V/E systems environment. With the change in architecture presented by HP's new 900 Series HP3000 systems and MPE XL, it is helpful to re-examine application design philosophies and techniques to insure optimum performance in the new environment.

Rather than attempt to present an exhaustive list of do's and don't's, a specific set of MPE XL features and techniques will be reviewed within the framework of improving application performance. Use of Native Mode (NM) versus Compatibility Mode (CM), extended addressing capabilities, and user mapped files will be examined. Coding examples and methods will be presented where appropriate to provide a background for discussion and elaboration.

With the understanding that application performance will generally improve as MPE XL tuning continues, relative performance comparisons will be presented to help quantify coding effort versus the resulting performance benefits.

Performance Opportunities

Because the 900 Series HP3000s are part of the overall commercial family of HP3000 systems, a great deal of attention was given to designing MPE XL for compatibility. The very high level of compatibility achieved has been well documented and has contributed to a constant stream of successful customer migrations.

At the same time, HP Precision Architecture (HPPA) offers a number of new opportunities for improved performance and productivity which go beyond the older HP3000 system capabilities. Since a number of these new features are used in the programming environment, it seems best to illustrate their power and benefits by actually going through a programming problem and the solutions available in MPE XL. This discussion will therefore follow the development of a program which will illustrate the benefits of NM versus CM, extended addressing capabilities, and user mapped file techniques.

The two essential goals are: 1) show *how* these new techniques are used, and 2) demonstrate that they provide *improved performance* over techniques used in the past.

The Problem

To facilitate comparisons of the various techniques, a single programming problem is desired which can be easily "modularized" in the solution. This allows leverage of common program procedures and makes it easy to "drop in" different techniques by simply changing specific program procedures.

A Table Lookup Simulation was chosen as a good problem to work on. Table lookup techniques are widely used in computer operating systems and applications and the design center is usually *fast and efficient data retrieval*, or in other words, high performance. In many applications, the requests for retrieval of entries are spaced randomly within the table and the table itself may be fairly large. These attributes tend to work against the requirement for high performance (as far as the typical operating system is concerned), so a table lookup simulation seems particularly good for determining the real power of MPE XL and the specific features to be examined.

Figure 1 shows a graphical look at the programming problem being posed.

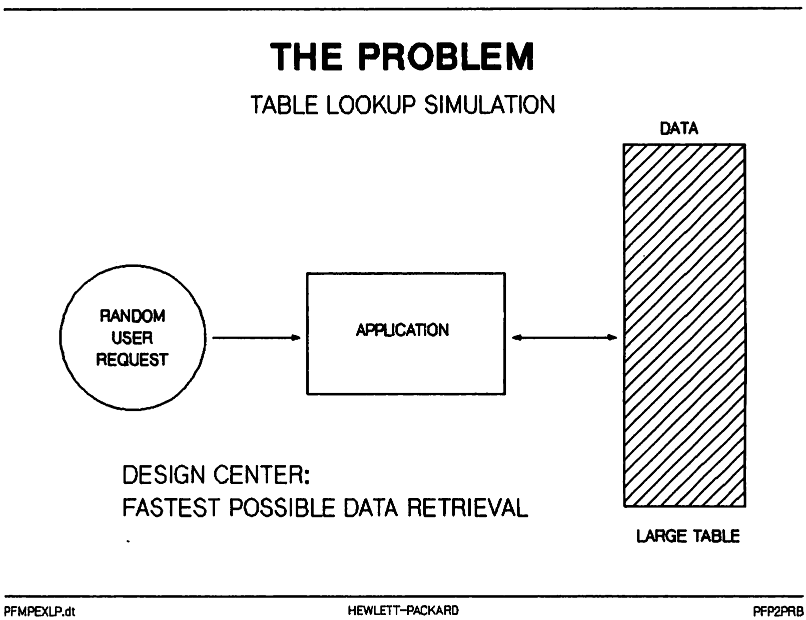


Figure 1

The design parameters for the Table Lookup Simulation program can be summarized as follows:

- The table itself will be very large and must be at least several times larger than can be represented in a single data segment on MPE V/E.

- Table entry retrieval must be extremely fast and efficient. It is desirable that the table be accessed as a memory resident structure.

- During the simulation, random entries in the table will be accessed.

- The program and associated files will be designed so that identical simulations can be run, each showing a different programming technique to do the actual table lookup.

- To gain a good performance comparison of the various techniques, a large number of table entries will be accessed during the simulation.

The Solutions

For this programming problem, there is really only one solution in the MPE V/E environment--use Extra Data Segments (XDS) to represent the table. (Note: Throughout this paper, it is assumed that a basic understanding of the Extra Data Segments capability of MPE V/E and MPE XL already exists. Our purpose here is not to show how to use the XDS intrinsics but how to use the new MPE XL features which essentially replace those intrinsics in functionality.)

In the MPE XL environment, we have four solutions to choose from:

- Table is multiple XDSs (program in CM).

- Table is multiple XDSs (program in NM).

- Table is a large array (program in NM).

- Table is a user mapped file (program in NM).

To duplicate the progression that a current MPE V/E programmer might take in migrating an existing application, we will first examine an implementation of the solution program using Extra Data Segments, then change the necessary procedures to use a large array (extended addressing), then finally change the program once again to define the table as a user mapped file. Performance improvement at each step will be noted and summarized at the conclusion.

NOTE

> To illustrate the various coding techniques being discussed, Pascal code fragments will be shown and referenced. These fragments are taken from a fully tested and executable program, however code which is not germane to the discussion has been left out to improve clarity. At the end of this paper is a complete listing of the actual Pascal program used for the "mapped file" version of the Table Lookup Simulation.

The Test Environment

All tests were run on a Series 930 configured with 64MB of memory and 4-7937 disc drives. The MPE XL version was A.01.10. To provide consistency in the performance comparisons, each test was run in a dedicated batch job environment which included the actual simulation itself and performance data collection programs. Each run used exactly the same data and script files (described below). Performance data was collected with XLDCP and AMT for a 15 minute period for each test. The essential performance indicators for the simulation were defined as follows:

- Elapsed time for the simulation.
- CPU time used during the simulation.
- Switch rate during the simulation (from/to NM/CM).
- % of CPU time in NM.

The elapsed time and CPU time measurements were built into the simulation itself using the TIMER and PROCTIME intrinsics. Switch rate and % of CPU time in NM were taken from XLDCP and AMT logfiles. All graphs shown here were generated from the XLDCP logfiles.

Data Structures

Before we can begin discussing the first programming example, the essential data structures need to be defined.

The table itself is defined as having 65536 entries, each entry 16 bytes in length. (Each entry could also be described as 4 MPE XL words, 8 MPE XL halfwords, or 8 MPE V/E words. The term *halfword* is used occasionally in MPE XL reference manuals to describe a 16 bit entity, as opposed to *word* which describes a 32 bit entity.)

The table entries will be accessed by index range 0..65535. With 65536 entries, each 16 bytes in size, we therefore have a table which is 1,048,576 bytes (or 1MB) in size. This is obviously quite a bit larger than can be represented in any single data structure in MPE V/E.

For the purposes of this simulation, each entry in the table is very simply defined (a real application would of course include useful data in the table). Each entry consists of two subfields: 1) A 32 bit integer value in the first word which is defined as the index number for that entry, and 2) The "DASCII" version of subfield 1 in the remaining 3 words (left justified). It will become apparent later in this discussion why this data format was chosen. Figure 2 below shows the table structure.

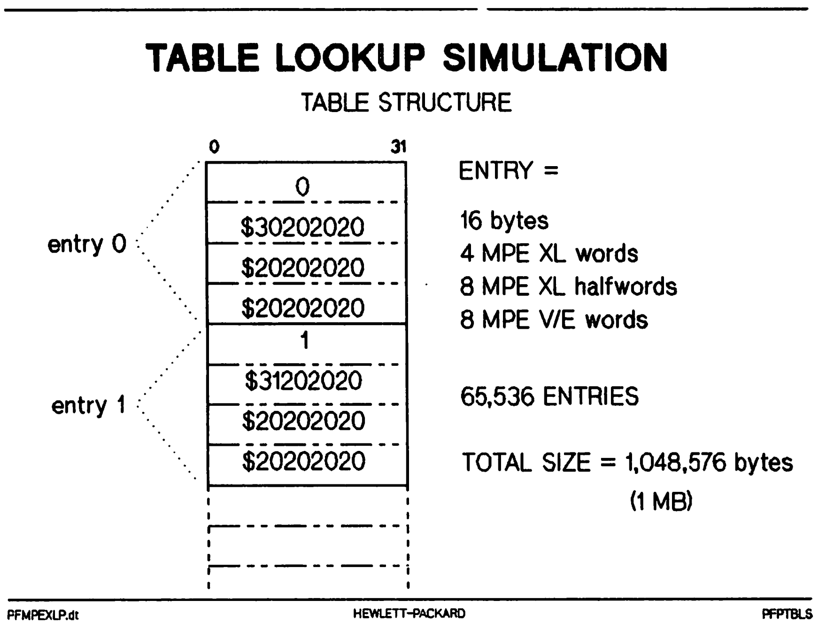


Figure 2

In our Pascal program we can define the table data structure as follows:

```
TYPE
  table_entry_type = record
    f1 : integer;
    f2 : packed array [1..12] of char;
  end;
```

Since one objective of the simulation is to access the table in memory, in the first two program examples (XDS and large array) it will be necessary to load the table from a flat data file on disc before the actual simulation of table lookups can begin. This is of course done only once during the program at the very beginning. A separate program was written to load the flat data file with the table data as described above. Our Pascal program would define the table entry and the source data file as follows:

```
VAR
  table_entry : table_entry_type;
  table_file : file of table_entry_type;
```

To simulate accessing the table, a large script file was created. Each record in this script file is an integer which has a randomly distributed value in the range 0..65535. As this file is read sequentially, each record's value will be used as the index for finding and processing the indicated entry in the table. (It should be noted that this scheme does not provide for any locality in the random table lookup. Most computer applications would exhibit at least minimal locality in this kind of data retrieval.)

The script file (hereafter called the *request file*) was created with 300,000 records to insure a good steady state simulation run that would last at least several minutes. Figure 3 illustrates the structure of the request file.

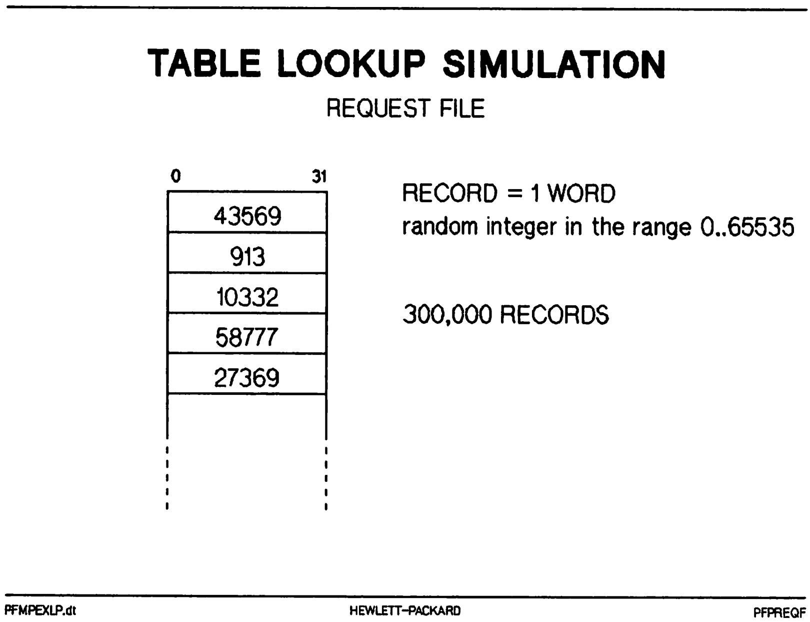


Figure 3

Table Lookup Simulation Using XDS

The data flow for this version of our program is shown in figure 4 below. Step 1 will be to create the Extra Data Segments required to hold the table. To represent a 1MB table in memory, 32 XDSs will be created, each 32768 bytes in size. Since each table entry is 16 bytes, each XDS will therefore hold 2048 entries.

In Step 2, we load the table with the data as defined above from an already existing flat file on disc (the *table file*). Once this is done, we are then ready to begin the simulation by starting a loop of reading the request file to get the table index requested, looking up the requested entry in the table, and then processing the entry (Step 3).

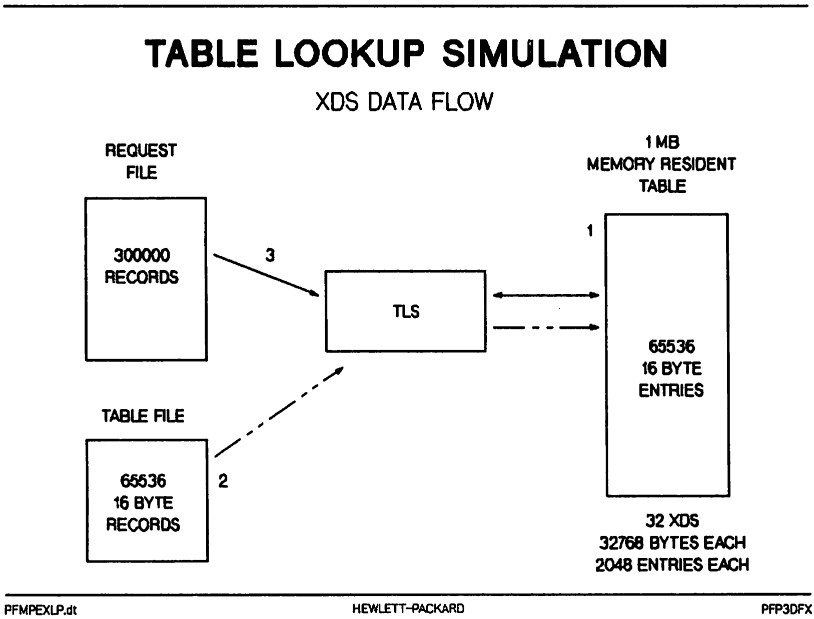


Figure 4

The commands and simulation output for the XDS test are shown below. User input is highlighted. The TLSXDSN program was compiled in native mode with the Pascal/XL compiler.

```
:FILE TABLE=TABLE.PROJECTS.TROUT
:FILE REQUEST=REQ1.PROJECTS.TROUT
:RUN TLSXDSN

Extra Data Segments created.
Loading the Extra Data Segments...
Table loaded.  Number of entries =  65536.
Request script file REQ1.PROJECTS.TROUT opened.
Starting Table Lookup Simulation...

Table Lookup Simulation completed on 300000 requests.
CPU time used =        597252 milliseconds.
Elapsed time of simulation =       610966 milliseconds.

END OF PROGRAM
:
```

As mentioned above, the design of the simulation program is structured so that each technique can be tested by simply replacing specific program procedures. The main body code for the XDS version of the Table Lookup Simulation program is as follows:

```
BEGIN  { main body }
   initialize;
  { start measurements }
   load_table;
   setup_loop;
   repeat
     get_entry_request;
     look_it_up;           { required only in the XDS version }
     process_entry;
   until no_more_requests;
  { end measurements }
   close_down;
END.
```

For the "large array" and "mapped file" versions of the simulation program, the only difference in the main body code is that there is no **look_it_up** procedure. The **look_it_up** procedure is required in the XDS example because we are managing a 1MB memory table in 32 Extra Data Segments. For each entry request, it is necessary to first determine which XDS the requested entry is in, and then calculate the offset within that XDS for the beginning of the actual entry.

This extra overhead is compounded by the fact that once we locate the entry, we must then move the data from the XDS to our user stack so that it can be processed. Already we can see that, aside from performance issues, use of XDSs is not very productive for the programmer when compared to using a simple large array or a user mapped file.

The code for **load_table** in our XDS program would look like this:

```
BEGIN  { load_table }
  xds_size := 16384;
  for dirc_ptr := 1 to 32 do
    getdseg (xds_dirc[dirc_ptr], xds_size, xds_id);
  reset (table_file);
  load_counter := 0;
  for dirc_ptr := 1 to 32 do
    for xds_entry := 1 to 2048 do
      begin
        read (table_file, table_entry);
        disp := xds_entry * 8 - 8;
        dmovout (xds_dirc[dirc_ptr], disp, 8, table_entry);
        load_counter := load_counter + 1;
      end;
  close (table_file);
END;
```

The **look_it_up** procedure would be coded as follows:

```
{ table_index has just been read from the request file }

BEGIN  { look_it_up }
  dirc_ptr := trunc (table_index / 2048) + 1;
  disp := (table_index mod 2048) * 8;
  dmovin (xds_dirc[dirc_ptr], disp, 8, table_entry);
END;
```

In this simulation, we are mainly concerned with the performance of retrieving a large number of table entries. What we do with the entry once retrieved is not really of interest. Of course, the bulk of a typical application would be in the processing of table data, not in retrieving it.

For the purposes of this simulation testing, our **process_entry** procedure is very simple. To verify that we have the correct entry, a simple comparison is made between the table index used to find the entry (from the request file) and the first subfield in the entry itself. From our description above of the table data, we know that the first subfield is nothing more than the index of the entry. This means that *our comparison should always show equal values.* If not, we have somehow retrieved the wrong entry!!

So, the **process_entry** procedure will look something like this:

```
{ table_index has just been read from the request file }

BEGIN  {process_entry }
  if table_index <> table_entry.f1 then
    { error };
  req_counter := req_counter + 1;
END;
```

The other routines in the Table Lookup Simulation program need not be examined in detail since they essentially do housekeeping, initializations, etc. Refer to the complete program listing at the end of this paper.

From the simulation output given above, we know that the XDS version of our program ran for about 10 minutes (610966 milliseconds to be exact). A graphical look at what happened on the system during the test is seen in figure 5. This graph shows CPU utilization during the simulation. (Note: In all test runs, the performance data collections were started two minutes before the simulation program was started so that the effects of the simulation could be clearly seen.) As expected, the dedicated batch job environment produced 100% CPU utilization during the simulation.

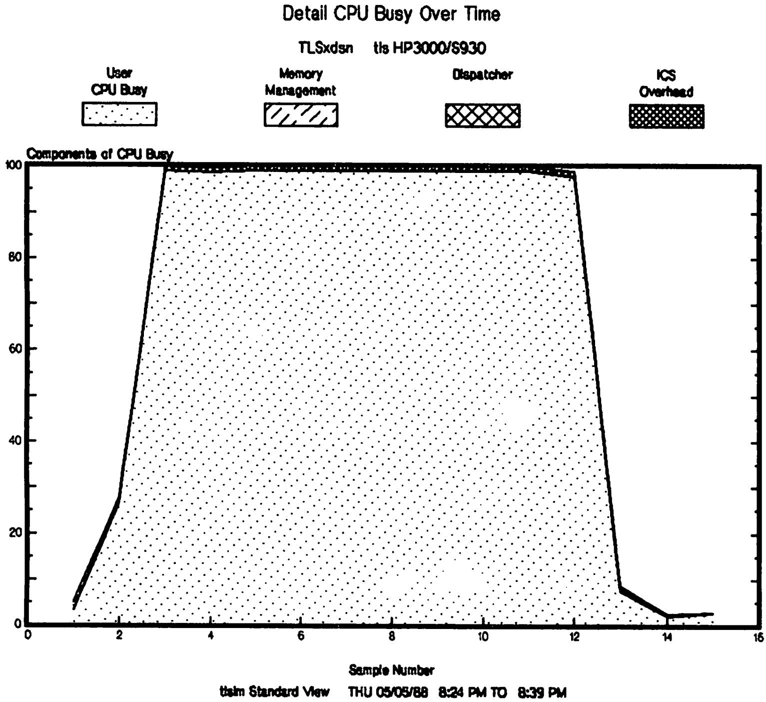


Figure 5

The really significant performance factor for the XDS example, however, is shown in the following chart. Because the XDS intrinsics reside in the compatibility mode SL (SL.PUB.SYS), a program running in native mode (as this one was) must access these intrinsics through the Switch Subsystem in MPE XL. Using the intrinsics is transparent to the native mode program because the necessary switch stubs already exist.

Figure 6 shows that during the simulation, the switch rate was sustained at about 600 switches per second. This is an extremely high switch rate and definitely contributes to degraded performance in the XDS version of the simulation.

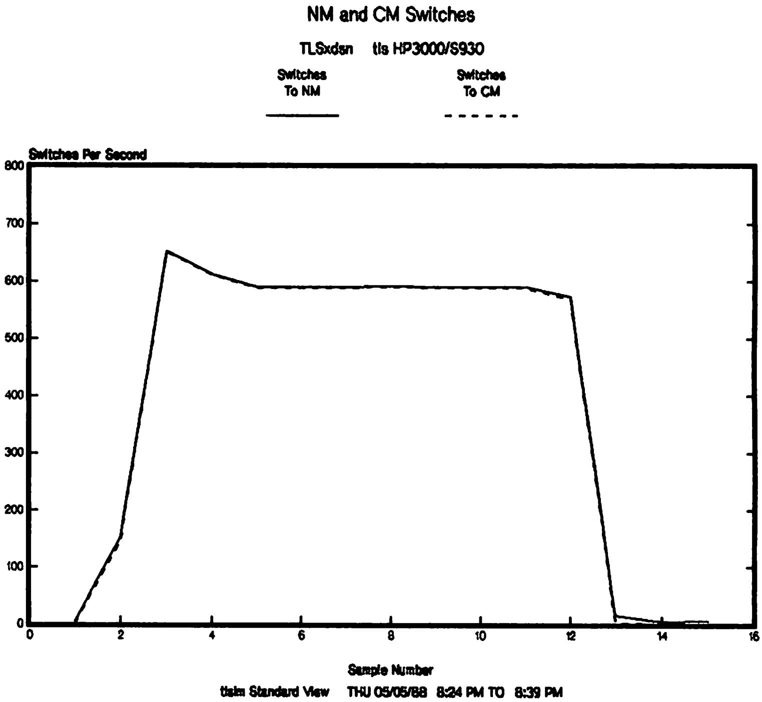


Figure 6

NM vs CM

To compare NM versus CM, the XDS version of the simulation was compiled using the CM Pascal/3000 compiler on the Series 930 and rerun in CM. In this case, elapsed time was even longer: 893342 milliseconds, or almost 15 minutes. The longer run time can be attributed to the overhead of CM versus NM for this kind of CPU intensive program and the fact that the number of switches was even higher in CM (see the summary table in Figure 12 at the end of the paper). Remember that the simulation issues 300000 sequential reads to the request file; the file system code for doing these reads is in NM, so the CM program must switch to NM for all the request file I/O.

As this test illustrates, NM performance is better than CM performance, even when part of the application is calling system routines which are in CM. This will generally be true for most applications being migrated to MPE XL, although there may be certain corner cases where it may be desirable to leave an application in CM (at least for now).

Table Lookup Simulation Using A Large Array

Now that a "past technology" solution for the Table Lookup Simulation problem has been established, the newer techniques which provide easier design and better performance can be examined.

With the extremely large addressing capability of MPE XL, it should be obvious that the table can simply be represented as a large memory resident array. Instead of having to create and manage multiple Extra Data Segments, a single block of memory space can be created which will contain the entire table. Indexing to a given entry in the table will be greatly simplified from a programming point of view and the XDS overhead is totally eliminated.

However, once the table array is created, there is still a need to load it from a disc file as in the XDS example. Figure 7 shows the data flow in the "large array" version of the simulation.

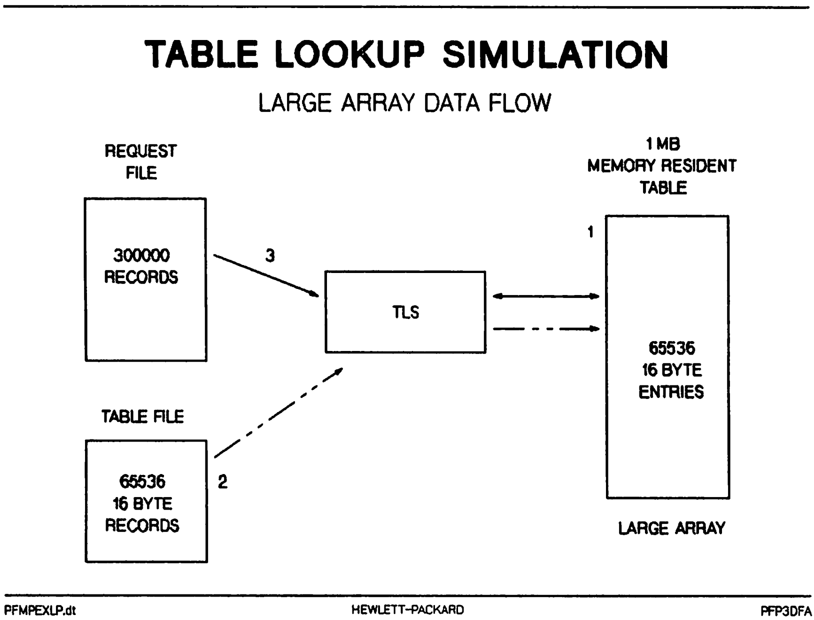


Figure 7

Although the data flow is essentially identical to the XDS example (refer to the earlier discussion of the steps involved), several procedures in the program will need to change. The main body code (refer to the listing given earlier) is the same except that there is no **look_it_up** procedure. The simulation test for the large array example resulted in the following output. Program TLSARRN was compiled in native mode with the Pascal/XL compiler.

```
:FILE TABLE=TABLE.PROJECTS.TROUT
:FILE REQUEST=REQ1.PROJECTS.TROUT
:RUN TLSARRN

Memory array created.
Loading the memory array...
Table loaded.  Number of entries =  65536.
Request script file REQ1.PROJECTS.TROUT opened.
Starting Table Lookup Simulation...

Table Lookup Simulation completed on 300000 requests.
CPU time used =        143231 milliseconds.
Elapsed time of simulation =       145824 milliseconds.

END OF PROGRAM
:
```

The code fragment below shows how the array is defined and also the declaration for the pointer which will be used to index the table.

```
TYPE
  table_entry_type = record
    f1 : integer;
    f2 : packed array [1..12] of char;
  end;
  table_type = array [0..65535] of table_entry_type;

VAR
  table : ^table_type;     { table pointer }
```

It would also be possible to index the table with a simple integer variable "pointer" which would be calculated each time an entry is desired. There is really no need to do this, however, since Pascal offers the pointer data type which can be conveniently used in the program syntax. This will be illustrated below.

One procedure which will obviously need to change is **load_table**. Instead of creating and loading 32 Extra Data Segments, the array can be created on the Pascal heap with the *new* function and then loaded by dereferencing the table pointer:

```
BEGIN  { load_table }
  new (table);             { allocates the array on the heap }
  reset (table_file);      { open the table data file on disc }
  table_index := 0;
  repeat
    read (table_file, table^[table_index]);
    table_index := table_index + 1;
  until eof (table_file);
  close (table_file);
END;
```

Notice how convenient it is to dereference the pointer `table` and specify an array element (`table_index`) in the read statement above. In one Pascal statement, we have read the table file and loaded data into the appropriate table entry in the memory array.

The only other procedure which needs to change is **process_entry**. Only one statement change is required:

```
{ table_index has just been read from the request file }

BEGIN  { process_entry }
  if table_index <> table^[table_index].f1 then
    { error };
  req_counter := req_counter + 1;
END;
```

Again, the power of Pascal syntax is evident in the `if` statement above. The pointer `table` is dereferenced by `table_index` and then the subfield variable `f1` which "retrieves" the correct table entry and subfield. In one statement, the table lookup is accomplished.

This version of the Table Lookup Simulation exhibited much improved performance over the XDS version; the elapsed time of this run (145824 milliseconds, or about 2.5 minutes) is a significant *4.2 times improvement* over the XDS test run! (Refer to the summary table at the end of this paper.) The major factor contributing to the improved performance is the almost non-existent switch rate since the XDS intrinsics are no longer being used. In addition, the path length of the table lookup is greatly shortened since there is no need to calculate XDS pointers and offsets. Using a large array provides for a much simpler and more efficient means of accessing the table data.

Clearly, using the extended addressing capability of MPE XL can result in impressive improvements in run-time performance in addition to the programmer productivity enhancement already mentioned. Can these great results be improved even more? Yes, they can.

Table Lookup Simulation Using A Mapped File

One part of the simulation that could be eliminated is the table loading. This is essentially "wasted" time in the simulation and needs to be done each time the program is run (in the XDS and large array versions). It would be far more desirable in this kind of application to make the table *durable* from one run to the next and somehow access it from our program as a memory array. This "best of all worlds" solution can be implemented by using a *mapped file*.

User mapped I/O is a feature of MPE XL which is particularly unique and powerful. The essential attributes are:

- A method of accessing data from files using a virtual pointer.
- Accessed using HPFOPEN intrinsic specifying a long (64 bit) or short (32 bit) pointer.
- File "reads" and "writes" are accomplished at the level of LOAD and STORE machine instructions.
- File System buffering and overhead is bypassed; structure of the data is user defined; access files like memory, memory like files.
- Can be much faster than normal file access, especially for non-sequentially accessed files.

User mapped I/O is possible because of the basic design of HP Precision Architecture. All objects to be accessed in memory (including files) are *mapped* into a large *virtual address space*. When a file is opened, it is assigned a virtual address range which encompasses the first byte to the last. By opening the file in such a fashion as to return to the user a *virtual address pointer* which "points" to the first byte of the file, all data in the file can then be accessed by dereferencing the pointer.

As the file is referenced, it is brought into real memory in *pages* from secondary storage (disc). The essential components of this *virtual demand paging* scheme are shown in figure 8.

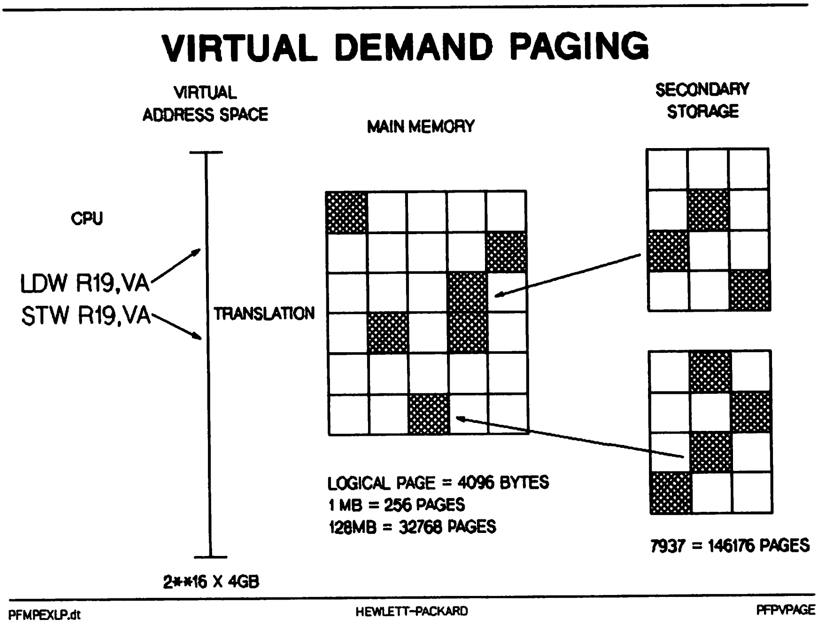


Figure 8

In the "mapped file" version of the simulation program, then, there will be no need to load the table data as a separate step. The memory table and the table file on disc (previously used to load from) will be *one and the same*. This greatly simplifies the data flow of the simulation, as shown in figure 9:

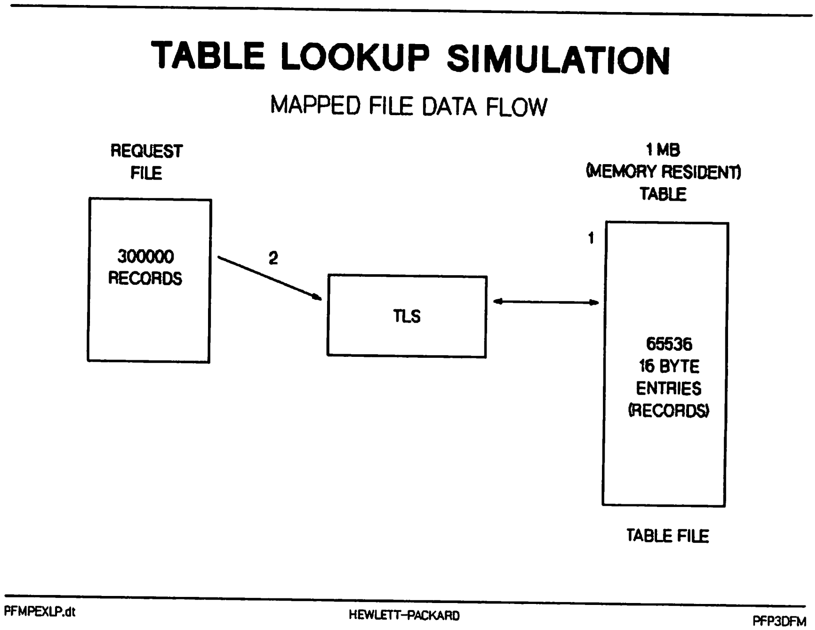


Figure 9

Only two steps are necessary: 1) Open the table file specifying mapped I/O, and 2) Loop through the request file. Although the entire table will not be memory resident at first, it will gradually become memory resident as more and more pages of the file are touched.

The simulation output for the "mapped file" test are shown below. The TLSMION program was compiled in native mode with the Pascal/XL compiler.

```
:FILE TABLE=TABLE.PROJECTS.TROUT
:FILE REQUEST=REQ1.PROJECTS.TROUT
:RUN TLSMION

Table file opened for mapped access.
Number of entries in table =  65536.
Request script file REQ1.PROJECTS.TROUT opened.
Starting Table Lookup Simulation...

Table Lookup Simulation completed on 300000 requests.
CPU time used =        116992 milliseconds.
Elapsed time of simulation =      124550 milliseconds.

END OF PROGRAM
:
```

As with the "large array" version, the main body code will be the same as the XDS version except that there is no **look_it_up** procedure (refer to the earlier listing). The **load_table** procedure will now become the place that the table file is opened for mapped access. No loading need be done, so the procedure is essentially just the HPFOPEN:

```
BEGIN  { load_table }
  hpfopen (filenum, status,
           ffd_option, table_filename,
           domain_option, permanent,
           access_type_option, read_only,
           short_mapped_option, table);  { return the pointer }
END;
```

Notice that the HPFOPEN intrinsic is called using *itemnum, item* pairs. This greatly improves coding accuracy and ease compared to the FOPEN intrinsic.

The important part of the HPFOPEN call is in the last line above; the *short mapped option* is requested and in the pointer variable `table` is to be returned the virtual address of the beginning of the file. Both the table array and the pointer `table` are declared in exactly the same way as they were previously for the "large array" version of the simulation (see code fragment above). As a result, the **load_table** procedure is the only piece of code that needs to change in order to convert the "large array" program into a "mapped file" program. Procedure **process_entry** is identical for the two versions (see code fragment above).

For a detailed look at the entire simulation and the coding technique for mapped files, refer to the complete listing of the program at the end of this paper.

From above, the elapsed time of the "mapped file" version of the simulation was 124550 milliseconds, or about 2 minutes. This is even more improvement over the XDS version than was seen with the large array technique, and results from not having to "load" the table as a first step in the simulation. To graphically see the benefits of the mapped I/O technique versus the XDS technique, compare the following graph with the earlier XDS graph showing CPU utilization:

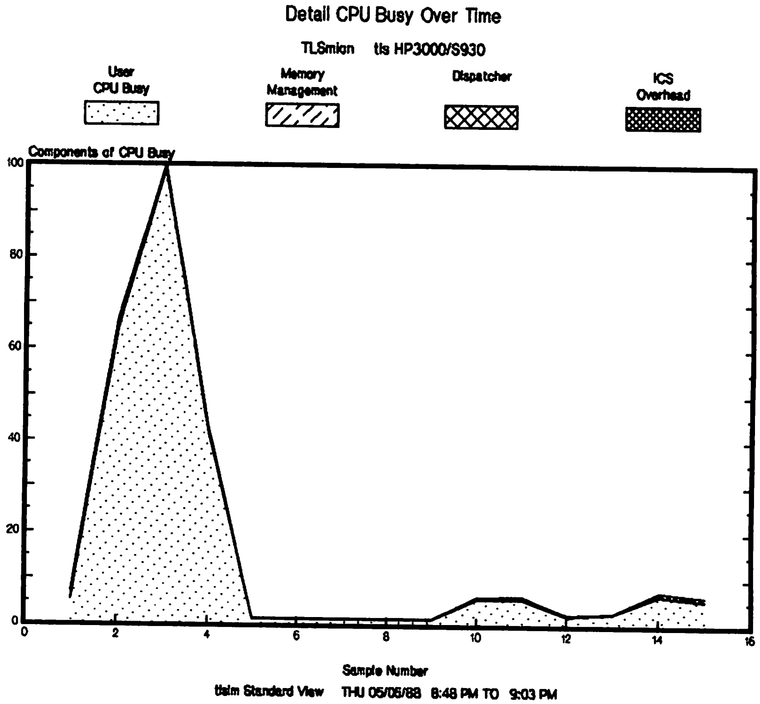


Figure 10

As expected, the switch rate dramatically improved over the XDS test. Note that the X-axis in the following graph is the same as before, however the Y-axis shows a much smaller range:

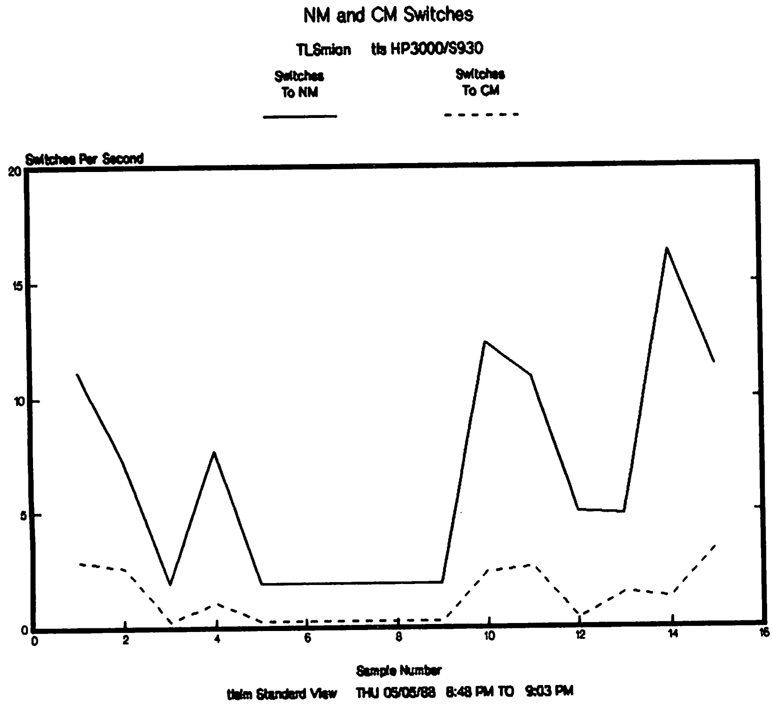


Figure 11

Mapped I/O Considerations

There are some considerations when using mapped files. Since the File System is being effectively bypassed, things like EOF and file posting are no longer being automatically managed. The user must explicitly use the FPOINT and FCONTROL intrinsics to set the EOF in those cases when records have been written beyond the existing EOF. This only need be done before the file is closed. (Since we are only *reading* the file in our example program there is no need for this.)

For critical applications, the user may want to use FCONTROL to force physical posting of file pages when appropriate, although too much of this would negate the benefits of using mapped files.

As noted above, mapped files may be opened with either a short pointer or a long pointer. For short pointer access, a file may be up to 4MB in size and a total of 6MB of mapped files may be opened at once per process. For long pointer access, a file may be up to 2GB in size and there is no limit to the total of mapped file space being utilized per process.

Short pointers are more efficient than long pointers and should be used wherever possible. If files are to be opened using a mix of techniques (FOPEN, HPFOPEN, HPFOPEN mapped, etc.) simultaneously, then use of long pointers will be required.

Conclusions

The following table shows the pertinent performance data for all of the simulation test runs. The "relative performance" column has been normalized to the XDS program running in native mode and is based on elapsed time of the simulation.

TABLE LOOKUP SIMULATION

RELATIVE PERFORMANCE
SERIES 930/64MB
MPE XL A.01.10

TEST	CPU ms	ELAPSED ms	NUMBER OF SWITCHES	% NM	RELATIVE PERFORMANCE
XDS, CM	872255	893342	931200	53.4	0.68
XDS, NM	597252	610966	731313	83.6	1
ARRAY, NM	143231	145824	112	99.9	4.2
MAPPED I/O, NM	116992	124550	110	99.9	4.9

PFMPEXLP.dt HEWLETT-PACKARD PFPRELP

Figure 12

The mapped I/O program shows the best results at 4.9 times the XDS program. The large array test also showed significant improvement at 4.2 times the XDS program. Both of these new MPE XL techniques illustrate the power and performance potential of HP Precision Architecture. In addition, it has been demonstrated that the programmer's task can be simplified and productivity improved. Clearly, the benefits of application design changes to take advantage of HP Precision Architecture are worth the effort.

Appendix A: Additional References

- Programmer's Skills Migration Guide (30367-90005)
- Accessing Files Programmer's Guide (32650-90017)
- MPE XL Intrinsics Reference Manual (32650-90028)
- *Hewlett-Packard Journal*, December 1987, "MPE XL: The Operating System for HP's Next Generation of Commercial Computer Systems"

Appendix B: TLSMION Program Listing

```
$OS 'MPEXL'$
$STANDARD_LEVEL 'HP_PASCAL'$
$CODE_OFFSETS ON$
$TABLES ON$
$VERSION '1.0'$
$TITLE 'Table Lookup Simulation Using Mapped I/O'$

PROGRAM tlsmio (input,output);

{ 1.0  05/02/88  Dave Trout, HP Rockville }

{ This program does a Table Lookup Simulation.  The purpose of this
  simulation is to compare the Extra Data Segments (XDS) capability
  of MPE for table handling versus Mapped I/O techniques and extended
  addressing capabilities in MPE XL.

  Data Structures:

    TABLE - 65536 entries, accessed by index 0..65535
    TABLE ENTRY - 16 bytes (4 MPE XL words|8 halfwords|8 MPE V/E words)

    Total table size is 1,048,576 bytes (1MB).

  The table will be accessed as a memory resident table using XDS
  intrinsics, a large array, or a mapped file.  To simulate accessing the
  table, a large script file has been created; each record in this script
  file is an integer which has a randomly distributed value in the range
  0..65535.  As this file is read sequentially, each record's value will
  be used as the index for finding and processing the indicated entry in
  the table.

  It should be noted that this scheme does not provide for any locality
  in the random table lookup.  Most computer applications would exhibit
  at least minimal locality in this kind of data retrieval.
```

```
  Since table lookup applications typically require extremely fast
  access, it is desirable to have an efficient access PATH to table
  entries which are MEMORY RESIDENT.  Performance of this simulation will
  be determined in large part on how well these objectives are met. }

CONST
  ccg = 0;
  ccl = 1;
  cce = 2;
  req_ffd = 'REQUEST';                        { request file }
  time_adj = 2073600000;                      { timer adjust }

TYPE
  xlstatus = record
    case integer of
      0 : (all : integer);
      1 : (info : shortint;                   { error number }
           subsys : shortint);                { subsystem number }
    end;

  table_entry_type = record
    f1 : integer;                             { logical entry number }
    f2 : packed array [1..12] of char;        { DASCII version of above }
    end;

  table_type = array [0..65535] of table_entry_type;

VAR
  status : xlstatus;
  no_more_requests : boolean;
  req_file : file of integer;
  table_index : integer;
  req_counter : integer;
  timer_start : integer;
  timer_stop : integer;
  time : integer;
  table : ^table_type;                        { table pointer }
  filenum : integer;
  cpu_time : integer;

FUNCTION timer : integer; intrinsic;
FUNCTION proctime : integer; intrinsic;
PROCEDURE hpfopen; intrinsic;
PROCEDURE fclose; intrinsic;
PROCEDURE ffileinfo; intrinsic;
PROCEDURE terminate; intrinsic;

PROCEDURE stop (parm : integer);

  BEGIN
    writeln ('*** Fatal error; parm = ',parm:5);
    terminate;
  END;
```

```
PROCEDURE initialize;

  BEGIN
    no_more_requests := false;
    req_counter := 0;
  END;

PROCEDURE load_table;

  { Since the table already exists as a permanent file on disc which
    will be opened and accessed using mapped I/O, there is no "loading"
    to do.  The table will become memory resident automatically as the
    simulation touches pages of the file.  This routine will HPFOPEN
    the file specifying the short mapped option.  The short pointer
    returned by HPFOPEN will then be used in later routines to access
    entries in the table. }

  CONST
    ffd_option = 2;                     { setup for HPFOPEN }
    domain_option = 3;
    short_mapped_option = 18;
    access_type_option = 11;
    file_eof = 10;

  VAR
    table_filename : packed array [1..10] of char;
    permanent : integer;
    read_only : integer;
    entry_count : integer;

  BEGIN
    table_filename := '%TABLE%';        { setup ffd }
    permanent := 1;                     { permanent file domain }
    read_only := 0;                     { read only access }

    hpfopen (filenum, status,           { open the table file }
             ffd_option, table_filename,
             domain_option, permanent,
             access_type_option, read_only,
             short_mapped_option, table);  { return the short pointer }

    if status.all <> 0 then
      begin
        writeln ('error on hpfopen; info = ',
                 status.info,', subsys = ',status.subsys);
        stop (6);
      end;
    writeln ('Table file opened for mapped access.');
    ffileinfo (filenum, file_eof, entry_count);
    if ccode <> cce then stop (7);
    writeln ('Number of entries in table = ',entry_count:6,'.');
  END;
```

```
PROCEDURE setup_loop;

  CONST
    afd = 1;

  VAR
    req_file_afd : packed array [1..28] of char;
    afd_string : string[28];
    n : integer;

  BEGIN
    reset (req_file,req_ffd);                { open the request file }
    ffileinfo (fnum (req_file), afd, req_file_afd);
    setstrlen (afd_string, 0);
    strwrite (afd_string, 1, n, req_file_afd);
    afd_string := strrtrim (afd_string);
    writeln ('Request script file ',afd_string,' opened.');
  END;

PROCEDURE get_entry_request;

  BEGIN
    read (req_file,table_index);
    if eof(req_file) then no_more_requests := true;
  END;

PROCEDURE process_entry;

  { Since we are using mapped I/O, we simply reference the entry we
    want to retrieve it.  For this simulation, just test to make sure
    we do indeed have the right table entry.  Since the first word of the
    table entry is an integer whose value is the table index, we can
    simply compare the index used with the retrieved value.  They should
    be the same! }

  BEGIN
    if table_index <> table^[table_index].f1 then
      begin
        writeln ('we have a problem here...index/entry don''t agree.');
        stop (4);
      end;
    req_counter := req_counter + 1;          { this one is done }
  END;

PROCEDURE close_down;

  BEGIN
    close (req_file);                        { close the request file }
    fclose (filenum,0,0);                    { close the table file }
    if ccode <> cce then stop (8);
  END;
```

```
BEGIN {Table Lookup Simulation}

  initialize;
  cpu_time := proctime;                       { start measurements }
  timer_start := timer;
  load_table;
  setup_loop;
  writeln ('Starting Table Lookup Simulation...');

  repeat
    get_entry_request;
    process_entry;
  until no_more_requests;

  cpu_time := proctime - cpu_time;          { end measurements }
  timer_stop := timer;
  time := timer_stop - timer_start;
  if time < 0 then time := time + time_adj; { fix time if required }
  writeln;
  writeln ('Table Lookup Simulation completed on ',
           req_counter:6,' requests.');
  writeln ('CPU time used = ',cpu_time,' milliseconds.');
  writeln ('Elapsed time of simulation = ',time,' milliseconds.');

  close_down;

END {Table Lookup Simulation}.
```

Effective Implementation of Distributed Electronic Time Management and Scheduling

John Ramuta
Hewlett-Packard Company
3410 Central Expressway
Santa Clara, CA 95051

Introduction - Time Management Systems

Time management systems are a significant tool to a great many people. How many times have you had someone say to you "Wait, let me check my calendar"? Indeed, some individuals would not be able to function properly without their trusty time management tools at their sides.

The popularity of time management tools is apparent by the proliferation of different systems that allow you not only to manage your time, but your phone numbers, finances, goals, contacts, even your diet. You can do all of this by the minute, hour, day, week, month, year, or longer. Several companies have even gone so far as to offer versions for children - as a way for them to "get a head start" on other children.

Time management systems have spread to the personal computer world as well. While computerized scheduling systems are nothing new, their use on PC systems has become more widespread than they ever were on mainframes, probably due to the "personal" nature of the desktop computer.

Time management systems of both kinds are much easier to find today. For the paper-based systems, there was a time when you either had to order them directly from the manufacturer or pick them up at an office supply house. Now you can get them in your local department stores, drug stores, or grocery stores. Electronic systems differ in size and range almost as much as manual ones do - ranging from PC-based packages such as Borland's SideKick, Timeworks Partner PC and the Microsoft Windows calendar to centralized packages such as HP Desk and PROFS.

Paper-based time management systems are not without their problems. For one thing, it's hard to get people to agree on the "right" way to do things, so there are multiple formats available, most of them incompatible in some way with one another. It is also difficult to schedule group activities, as the availability of the workgroup members is not stored in a central location. The size of some of the systems has caused a number of people to make use of more than one system: One for the workplace, and another, simpler version for evenings and weekends. But this also complicates the process because appointments may have to be transferred between systems.

Electronic time management systems also have problems. First, you can't take them with you very easily. Second, most people have been using some kind of paper-based appointment book for some time, and don't want to change. Third, if the electronic scheduling package runs on an individual's PC, then it is difficult (if not impossible) to perform any kind of group scheduling.

To some extent, the problem of portability is being addressed through the use of portable computers and specialized "calculators" that track appointments. However, this does not provide a solution for

An "Ideal" Electronic Calendaring System

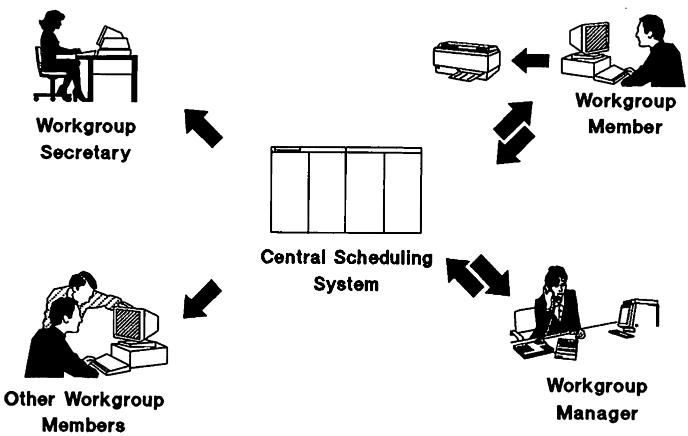


Figure 1.

the person who wishes to use a paper-based system. And the portable computer also fails to address the group scheduling problem.

An Ideal System

A combination of both paper and electronic systems best suits the needs of the workgroup *(figure 1)*. A "centralized" electronic system can contain all appointments for the people in a workgroup. This places the necessary information in one location accessible in various degrees to everyone in the group. Workgroup members maintain their own calendars on this central system, either via direct terminal-based entry, or via their personal computer, which can upload that information to the central system.

This upload process eliminates the need for the workgroup member to learn how to use the central system. He can go on using the calendar program that he is used to using, eg., SideKick. The information can be moved to the central system via a "batch" process, probably at a time when the user is not present.

The department manager or his designate can schedule group events on the central system, using the input from all members of the group. These group meetings can be added to the schedules of the workgroup members. The department manager can also use this information to control the resources of the department. Scheduling information can easily be moved into a spreadsheet or project planning package.

Members of the group are able to look at the schedules of other members through some kind of simple "Whereis" command to find out where they are or what their future schedules look like. This helps in simplifying job coverage when a member of the group is away, as well as eliminating the need for a group member to "let everyone know" where he is. That information is already in a known location.

Each member of the group can have a "personalized" system that ties into the main system in some way. This can be a paper-based system, or it can be electronic. The workgroup member can even elect to have both. This personal system uses a format familiar and comfortable to the the group member. For instance, if someone is using a particular paper-based time manager before implementation of the electronic system, he can go on using it. The central system just generates hard copy in the old format. The workgroup member could continue to use his existing PC-level time manager (if any) or could learn how to enter appointments directly on the central system. The workgroup member could learn about the additional capabilities of the new system (alternative output formats, uploading and downloading to PCs, etc.) as necessary.

The system must be expandable. This can be accomplished by modularizing the system very carefully, and allowing for "hooks" to other packages to be inserted at a later time. Such interfaces between modules must be carefully documented and rigidly adhered to.

The HP Desk "Engine"

The calendar section of HP Desk is the type of centralized system that serves these needs well. Workgroup members can easily store their appointment information on this system. The group manager can retrieve that information and use HP Schedule to set up group meetings. Using HP Desk as a basic "engine" and adding software customized to the workgroup's particular needs provides a powerful combination for time management activities.

CALSCAN

CALSCAN was originally conceived as an HP 3000 program to address some of the issues raised earlier in this paper. It scanned through the HP Desk calendar for a particular user, and returned a sequential ASCII file containing the requested output. This output could then be used by a number of different programs for various purposes, or printed directly. The HP 3000 version of CALSCAN became the central module of a series of scripts and programs that allowed users to look at or change information contained in the HP Desk database. For example, it is used within a package that allows schedules to be moved between HP Desk and the Time Manager program on the HP Portable Plus computer. Another set of routines implements a "Whereis" function within HP Desk itself.

In its original form, CALSCAN did not take much advantage of the processing power of the PC. The program was HP 3000-driven. To take advantage of the idle processing power, it was decided to

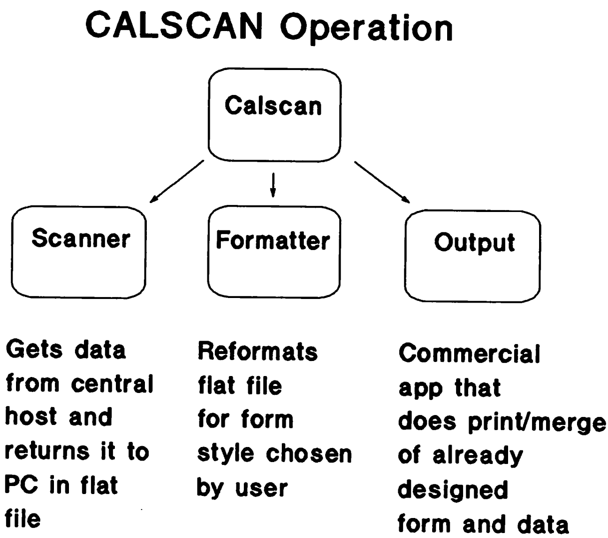


Figure 2.

move most of CALSCAN to the PC. This would also give us a greater flexibility in output formats, as well as making it easy to move the data into other PC-based applications. Because as "robust" a package as possible was wanted, it was decided that custom code was the proper way to go, rather than trying to rely on a combination of command files, scripts and macros to do the job. It was also desired that the package appear to be "seamless", even though several programs might be called to accomplish all the steps. The application had to be available through either the HP New Wave environment or the standard MS-DOS command interpreter.

The new system had to accept data entry through the MS-DOS command line, via a full-screen data entry form, or by way of a Windows dialog box or similar construct. The modular approach enabled the alteration of programs (or the addition of new ones) without affecting the rest of the system. It also had to allow for the smooth transition to Windows (and later, to the OS/2 Presentation Manager).

It was also decided to incorporate outside packages and tools into the code to reduce the amount of time needed to complete the package. The initial pass on the package would be a non-MSWindows approach, and would be encapsulated into the New Wave environment as an "oldapp". Again, this would reduce the coding time.

For handling the non-Windows user interface, we chose an in-house input forms package. To handle the final output, we selected a third-party output forms package. To provide the interface between the PC and the HP 3000, we used HP Co-operative Services.

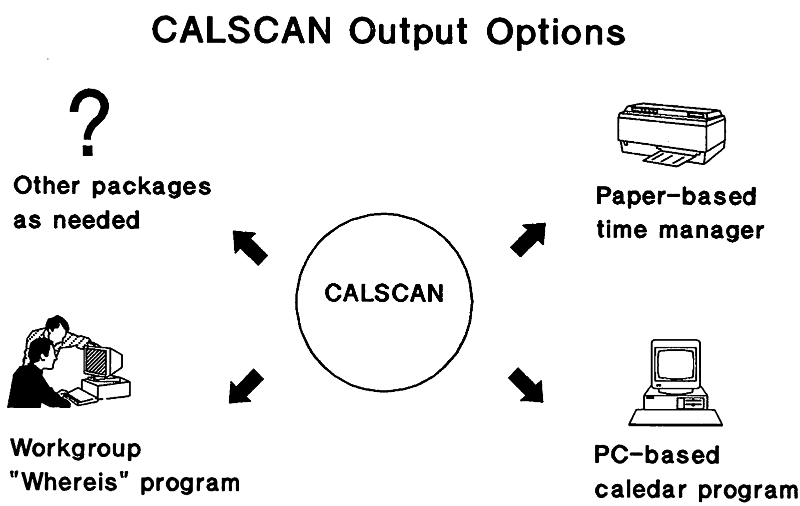


Figure 3.

HP Co-operative Services (HPCS) is a series of routines which can be linked to a PC application program to provide direct use of certain HP 3000 intrinsics within a PC-based program. These intrinsic routines can be used by PC programs written in COBOL, PASCAL, or C. (The PC part of CALSCAN is written entirely in Microsoft C, version 5.0).

In the final design *(figure 2)*, CALSCAN is the driver program, and the user interface. It, in turn, executes the other programs as necessary. The scanner program gets the information from the HP 3000, returning a flat file to the PC. The formatter program reformats that flat file according to the user's choice of output format. And the output program, a commercial application, performs a print/merge operation for the appropriate print device. The user is unaware that several programs are operating, and the main program is all that must be encapsulated into the NewWave environment.

Output from the CALSCAN program can be handled in a number of ways *(figure 3)*. The user can elect to send the output to a laser printer to get hard copy matching the format of his paper-based time management system. He can have it downloaded to his PC to use in a local calendar program, or to place the information into another PC package. The workgroup as a whole can have a "Whereis" program that would enable them to quickly query the HP Desk system as to a member's current location. This can incorporate security checks.

Calscan vs. "Ideal"

Compare this design to the "ideal" stated earlier. The system uses HP Desk as the central storage location for the schedules of everyone in the work group. Users enter their data either through the usual HP Desk interface, or into their PC-based calendar package. If HP Desk is being used, then no other steps need be taken. If a PC-based package is used, then the additional step of moving the data to the central system must be added. However, this upload of information could be automated to occur when the user is away from his desk.

Monday	Tuesday	Wednesday	Thursday	Friday
2 1200 PHONE DUTY	3	4	5	6 1330 Get Info off Ruenn
9 1200 Lunch w/Kristy and	10 1200 Lunch with Seth	11 1200 Phone Duty	12	13 0900 Division Coffee Br

Figure 4. A monthly format.

Each user can download his own scheduling information to the PC, where it can be printed in a desired format (figure 4) or moved into another program, for example, SideKick or Windows. The problem of mulitple calendars (one on the 3000 and one on paper) has been eliminated.

Terminal-based group scheduling is possible through HP Schedule. The group manager is able to schedule members of the group because the information to do so is all in one place. A PC-based group scheduling program can be written to fit into the CALSCAN system. For example, a program can be added that would download information for several members of the workgroup and format that information appropriately for a spreadsheet or project management package. This allows the group manager to make better use of his available resources. Figure 5 shows a sample Excel spreadsheet containing this "consolidated" information.

JUNE - 1988		Mon-30	Tue-31	Wed-1	Thu-2	Fri-3
Mike Smith	am	Training	Off-Site		Las Vegas	Las Vegas
	pm				Las Vegas	Las Vegas
Susan Jones	am					
	pm					Vacation
Neil Lambert	am		Sales Training			
	pm					
Patricia Evans	am	Vacation	Vacation	Vacation	Vacation	Vacation
	pm	Vacation	Vacation	Vacation	Vacation	Vacation

Figure 5. An Excel spreadsheet.

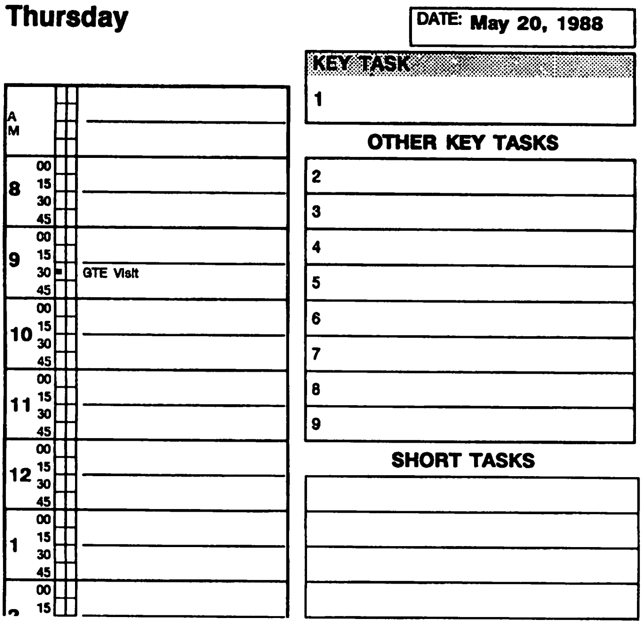


Figure 6. A daily format.

If a workgroup member's job responsibilities suddenly changed and the output format he was using no longer suited his needs, it would be a simple matter to change over to a different output style. Nothing else would have to be altered. For example, instead of the monthly output format shown in figure 4, he could use the daily format displayed in figure 6. Other formats could be easily customized using the formatter.

Workgroup members can check one another's schedule by requesting that one instead of their own. A "Whereis" facility of this sort would have to be simple to use and quick to run. It would not provide for any hardcopy output. It should probably run as a pop-up type window so that it can be accessed at any time.

Conclusion

Calendaring systems have become quite important to many people to allow for the proper management of their time. However, too many of these systems, electronic or otherwise, suffer from a lack of flexibility. This has limited their usefulness, and in a few cases made them unusable.

Most of the tools for implementing a flexible electronic system exist. It is a matter of combining the right off-the-shelf packages with some amount of customized code to create the kind of system that will be useful to the workgroup. The CALSCAN system discussed here is just one possibility, one that we expect to continue evolving as both new needs and new tools appear.

"I Have to Teach the Others Back at Work": When Customer Education Is Really Train-The-Trainer

Mary Humphrey
Hewlett Packard Co.

You are conducting the usual introductions to a full class of customer students. As you go around the room asking names, company or department, and reason for coming to class, you discover several students have come with their own "hidden performance objective." They were sent to training with the expectation that they would return to work to reteach the course to their coworkers. They may be apprehensive about this assignment, or they may feel it is an acceptable way to cut training costs for their company.

Problem or Opportunity?
This opening situation has the potential to produce students who feel threatened or disappointed, students' managers who perceive your training to be ineffective or failing to meet their needs, and "the others back at work" who will continue to be untrained and will not be good prospects for future training. Or, the situation could result in students who appreciate the efforts you make to help them achieve realistic results with reteaching, students' managers who perceive the value-added to your training, and the others back at work who will receive some benefit from reteaching and who will become candidates for future training. The key factor in determining which result occurs is the trainer.

The following list of trainer's DOs and DON'Ts is a prescription for turning a hidden performance objective into a positive result. It is important that none of these activities requires a change in course content or organization, none requires a special preparation by the trainer, and none compromises the delivery of the training to other students. In fact, several of the DOs can actually improve students' level of learning and the probability of transfer of training to on-the-job use.

[The conference presentation of the following list includes examples and demonstrations of specific techniques. There is a session handout of notes and the personal contract listed in item 7.]

1. DON'T Ignore Their Needs or Waste Time Trying to Change Their Objective
Ignoring a student's objective to be able to "teach the others back at work," is a poor idea. The objective won't just go away. It is often accompanied by a student's fear that he/she will not learn well enough to teach others. This can interfere with learning, increasing the probability that the reteaching will be a failure.

Another point to keep in mind when you encounter the hidden train-the-trainer objective is that it is a waste of time to argue that the objective is unfair to the student, a misuse of training, not highly successful...or any of a number of other truths. You are not going to change the mind of the manager who made the decision by presenting counter arguments to the student. Trying to do so is likely to further stress the student and reduce chances of successful learning even more.

2. DO Reset Their Expectations and Motivation
Restate the basic rule that knowing about how to do something is not as hard as actually doing it, but teaching how to do something is even more difficult. Set the expectation that being able to reteach the course material will require extra effort and detail on the students' part. Make clear that you will make changes in your presentation to help them, but that you can not guarantee that they will achieve their reteaching objective.

You can create a positive attitude by pointing out that the efforts and techniques they will use to help prepare themselves to reteach are also likely to increase their own learning and mastery of the material. Most people are aware that they recall very little of what they hear, a bit more of what they see, and even more of what they both see and hear. Continue this analysis by pointing out that the best levels of retention are when people say what they are doing as they do it. This is a technique they will be using when they prepare to reteach and when they actually do teach others. While their assignment to go back and teach the others may not be easy, it does have some benefits to them personally.

3. DO Use the Course Design
Start with the Student Performance Objectives (SPOs). Identify those that will be critical to reteach to others in order for them to achieve the overall course goal. Be sure the course goal and SPOs are visible to the students. Point out how the course design provides teaching activities to support the achievement of each SPO. State clearly what activities are planned for each SPO, and encourage students to make note of the connections. Knowing the purpose of each activity will help them to present the activities to others. It will also help to keep them organized and thorough.

4. DO Make Your Teaching Techniques Visible
At the beginning of every training activity, state how you will present it and monitor their learning. Tell the students key information such as why you have chosen a given method, what effect an example or demonstration is designed to have, how to make key information visible, and when to give practice and feedback. Remember that they probably will not be conducting a class, but tutoring 1:1 or with small groups. If appropriate, make suggestions for how they might adapt a presentation of the course content for a small group. Keep this brief and specific. You spend time better by modeling a useful teaching technique than by talking about how to do a diiferent one.

5. DO Stress the Use of Class Materials and Other Support Resources
Remind the students often to take detailed notes that will be useful to others. If the course materials include references, documentation or other written material, emphasize the students' need to be familiar with what information they contain and how they are organized. Identify other post-class resources such as coworkers who may have already had the same training, on-site support services, or management.

6. DO Present Practice and Feedback On Two Levels
To successfully reteach, the students will need the usual practice and feedback on performance, but they will also need to know how to give others practice and feedback. Simply asking them to restate the objectives and instructions for each lab or exercise will be useful. Ask them also to state how performance is measured and give examples of acceptable performance of the lab activity. Point out typical errors others might make in an exercise and give examples of how to monitor for this and give corrective feedback. Be sure students understand the purpose of a practice activity as well as how to deliver it -- they may need to adapt labs for use back at work.

7. DO Plan for On-the-Job Application Including the Reteaching
At key points in the training and at the conclusion of the class ask students to plan for their use of the information or skills back at work. With students who plan to reteach, you can present teaching others as a particularly effective post-class practice for themselves.

One good technique for improving their probability of success at reteaching is the Personal Contract Method. It requires only a pen and paper and about 10 to 15 minutes. Have students actually write out answers to each of the following questions. Review their answers and provide any additional answers, explanations, examples or other help to assist with the weak points they identify.

* Who will you be teaching and when (give specific names, dates, times if possible)?
* What Student Performance Objectives will you select as appropriate for your reteaching? (List or check off SPOs from a class materials list.)
* What teaching activities will you present to support each SPO? (List briefly, or circle items from the class agenda or table of contents.)
* What practice will you provide and how will you monitor performance?
* What will you need to practice, prepare or organize to be able to teach the others?
* Review your answers to the above questions. What is most likely to go wrong?
* What can you do now (before the end of class, or before you have to teach others) to strengthen the weak point or reduce its probability of occurring?

Asking students to sign their contracts, exchange copies with each other, give you a copy, or plan to review them with their managers, are all additional steps you can use to make the contracts more effective.

Conclusion
The next time a customer comes to your training and announces that he/she is expected to go back and teach the others at work, view it as an opportunity to make your teaching skills pay off for both of you. You can help the student achieve his/her personal objective while you enhance your own motivation and value as a trainer. And you can accomplish this with relatively little effort and planning.

PLANNING COMPUTER MAINTENANCE PROGRAM TO MEET YOUR NEEDS

SIAOU-SZE LIEN
Hewlett-Packard Singapore (Sales)
1150 Depot Road
Singapore 0410

INTRODUCTION

Today, computers touch nearly every aspect of our business. It is increasingly important that we safeguard the data contained in them. Loss of a single file can mean thousands of wasted dollars spent on recovery, not to mention the priceless nature of data that might be lost forever.

If you own or manage a computer center, you or your company would have already spent thousands of dollars on hardware equipment, data storage media and systems or applications software. However, to ensure a successful operation, one very important step must be taken. That step is the development of a comprehensive computer maintenance program for your facility.

This paper covers various aspects of computer maintenance that every systems manager should know about.

I will first outline the various maintenance requirements throughout the "life cycle" of a computer system and the cost implications of these components. I will discuss how to choose a computer systems maintenance program for both ongoing hardware and software support. I will also cover the financial analysis for computer maintenance. Lastly, I will discuss some future trends in computer maintenance systems.

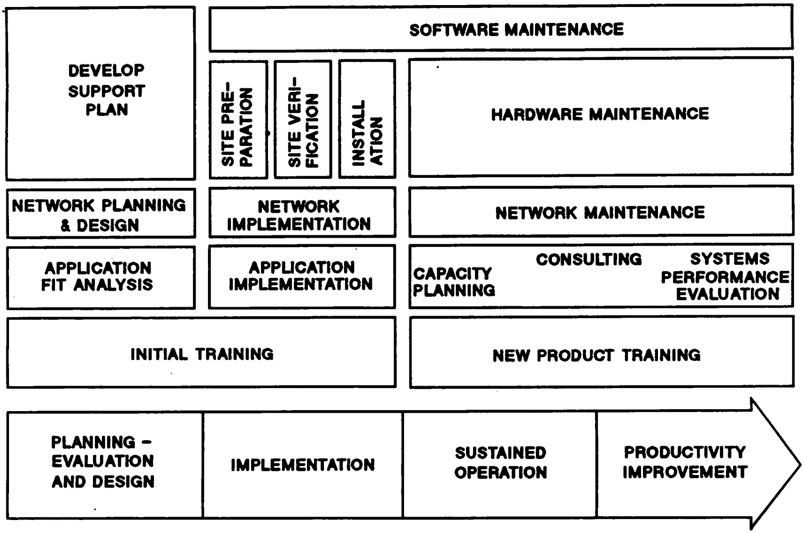
SYSTEM SUPPORT LIFE CYCLE

When one talks about computer maintenance, the first thing that comes to mind is hardware or software maintenance. In actual fact, there are other maintenance needs throughout the life cycle of a computer system implementation.

First, let's look at the process of implementing solutions for an organization. There are three phases involved:

1 Planning,
2 Implementing and
3 Operating.

PLANNING FOR YOUR SUCCESS: LAYING THE GROUND WORK



SYSTEM SUPPORT LIFE CYCLE

The planning phase entails designing and evaluating end-user systems needs, training, manpower planning and developing a system support maintenance plan, among others.

In the implementation phase, some of the key activities include system implementation and installation and end-user training.

On an ongoing basis, the operation would have to be maintained, fine-tuned and, in many cases, expanded to cater for changing business needs.

In each of these phases, there are associated maintenance requirements.

* Evaluation and Design Phase

During the initial phase of analysis and recommendation, you may want to rely on the training programs or consulting services to select the type of system that best suit your needs. These consulting services may also help define system requirements and software needs. They will facilitate the transition to a new system.

In the area of application, time may be needed to perform application fit analysis. At this stage, a decision would have to be made to determine whether the application requirements are going to be satisfied with an off-the-shelf package or in-house development. If the solution is in the application software package, the level of extra development effort required will have to be determined.

In the area of networks, planning and design take place. You need to select the best network strategy to support your business objectives. All data flow across multiple systems are analyzed and a detailed network design plan could then be determined. This early planning in your network life cycle ensures future flexibility, cost savings and a higher probability of success in implementing the network within your organization.

Developing a support plan for hardware and software maintenance is necessary to ensure that financial resources are dedicated for the right purpose. This support plan should reflect the organizational support needs in the areas of software and hardware maintenance. The criteria should also be made clear to facilitate the objective in selecting a system maintenance contract.

* Implementation Phase

Here, there are several specialized services to help put your implementation plan into action. Some of these services include site planning and preparation, verification and system installation.

In network implementation, the site preparation includes the installation, verification and testing of the necessary data communications and line equipment. This is followed by the execution of the network implementation plan devised in the design phase.

* The Operating Phase

Hardware and software maintenance services provided by the vendor are necessary to ensure the continued and smooth operation of the system. The user can be classified into two categories. In one category, the user essentially maintains status quo after installation. In the other category, the user will plan for expansion. The latter, of course, requires more effort. New product training would be desirable to keep up with the latest technology of the vendor or vendors. It is essential in the planning for expansion. Capacity planning is usually a good practice to manage the system requirements on a pro-active basis.

The overall picture we are seeing is a complex one. Overall system maintenance requires careful planning. Not only must the support requirements be taken into consideration, so must the cost be evaluated. The trade-off between cost and support level will have to be managed.

COMPONENTS OF COMPUTER SYSTEM MAINTENANCE COSTS

In looking at the maintenance costs for a computer system, there are several components. System costs are usually the most visible of the capital costs. This is only but the tip the iceberg. Some of the other costs are hidden which should be clearly spelled out. The user should be aware of them. The "hidden costs" usually represent between 50% to 130% of the total hardware costs. These "hidden costs" may include:

1 Software costs,
2 Training costs,
3 Site Installation and Preparation costs and
4 System Maintenance costs.

Consultancy services may not necessarily be included as part of the system acquisition costs.

The details of the cost components are as follows:

1 Software Costs

Software costs vary immensely with the system type and application. Mainframe applications software tend to be more costly than those on minicomputers, for example. Software costs can sometimes range up to 100% of the total system cost. There are two categories of software cost:

a) Application development cost and
b) Maintenance costs.

Application development software can be further divided into two types. They are off-the-shelf packages or in-house developed software. What is hidden in off-the-shelf packages may include the programmers' time to customize them to suit the end-user requirements. In-house software development involves the purchase of compilers and time to develop the applications. There is the additional factor of system analysis time which is not included anywhere else in the costing exercise. This cost can usually be fairly substantial.

Application maintenance is usually required from software supplier to cater for upgrades and enhancements. Program maintenance would be needed for customized packages and the software developed in-house. An additional item which can contribute to higher costs is the maintenance of the operating system.

2 Training Costs

Training is essential for the successful implementation of a computer system. This usually accounts for 5 to 10% of the total system cost. Initial training comes in the form of new product training to familiarize the users with the features of the new software and hardware. Only with a clear understanding of the system capabilities and limitations could one exploit the system to its fullest. In HP's terms, this would be translated into courses like Programmers' Introduction, Systems Manager and Operation.

On an ongoing basis, further education increases the productivity of the MIS staff. This usually comes in the form of more advanced courses. What in-depth knowledge really means is that the system can be further exploited to service the end-users. In addition, an MIS department should always cater for attrition.

3 Site Preparation and Installation Costs

In many cases, one factor which is often forgotten is the site preparation cost. Site preparation costs include the time your hardware engineer spends verifying and planning for your machine. It may in some cases include a designer fee to plan for the computer space, air-conditioning, electrical and other environmental requirements. Site preparation costs are borne by the customers to ensure that the specifications are met to cater for the computer system. With many vendors, site planning, verification and installation are usually included in the system cost. Site preparation costs typically run between 2 to 10 % of the total system costs.

4 System Maintenance Costs

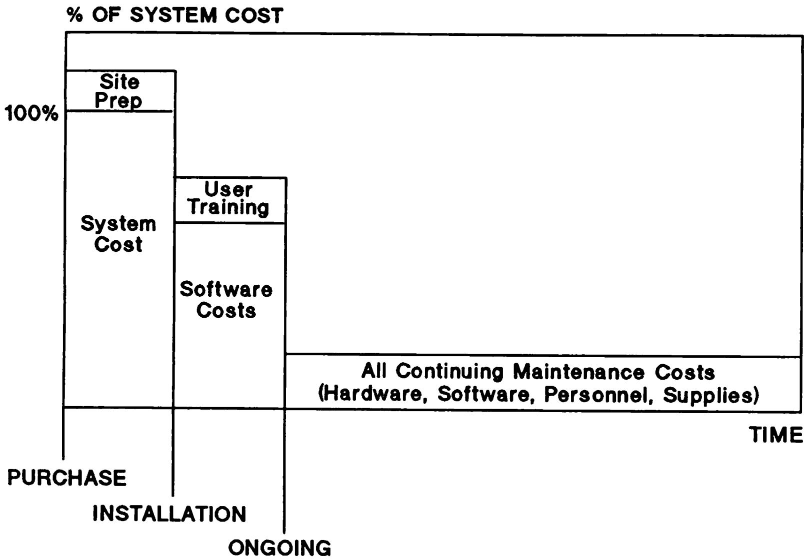
System maintenance costs, however, run between 10% to 20% of the system costs per year. It is a recurring cost necessary to maintain a highly available system. System maintenance can be broken into three components, i.e. hardware and software maintenance and consumables. Hardware maintenance costs typically ranges from 5% to 9% per annum. This, of course, is dependent on the system availability requirements and hardware configuration. Software maintenance costs also vary with the software type and level of support service. Computer consumables cost depends on the usage of the peripherals like magnetic tape, disc drives and printer.

On an overall basis, one can see that the cost components of the total system vary greatly. The total solution cost in the first year, if we include all the hidden costs, would be about 37% to 140% more than the system cost. The total costs can be broken down into capital expenses and recurring expenses:

Category	% of Total System Cost	Cost Type Capital	Cost Type Recurring
System costs	100	X	
Software Costs	20 - 100	X	X
User Training Costs	5 - 10		X
Site Preparation Cost	2 - 10	X	
System Maintenance Costs	10 - 20		X
Total Solution Cost	137 - 240% (of system cost).		

Obviously, not all the total solution costs are incurred at the same time. Typically, the organization will incur the system costs and site preparation cost at the point of purchase, and software and user training costs during installation. All maintenance for hardware, system and application software, personnel and consumables are incurred on an ongoing basis.

COST FLOW TIMING ANALYSIS



To summarize what we have discussed so far, good planning and a balanced investment are needed to ensure that the system is utilized optimally with well-trained MIS staff and satisfied users.

COMPUTER SYSTEM MAINTENANCE PROGRAM SELECTION CRITERIA

Selecting a maintenance program requires some careful thoughts in order to ensure the right support services meet the needs of the organization. The bottom line, of course, is price/performance.

Before we delve into the selection criteria, it is necessary to understand the hardware and software maintenance needs of a computer system.

Hardware Maintenance Needs

There are several features in a hardware maintenance service that the user will require. One of the key features is system availability. To achieve this, the maintenance service must be able to provide a mechanism whereby the user can obtain quick resolution to systems problems which may impair their operation.

At the same time, some hardware components are expected to be upgraded to improve the system performance or have bugs removed. It is also expected that such engineering changes and improvements will maintain system compatibility. Engineering changes are also needed to increase the system functionality. It is desirable to have further enhancements to the system so that the capabilities can be increased over the years.

The ability to predict system or component failures before they occur is a highly desired feature. This allows the user and engineers to take pro-active measures to alleviate any potential problems. What this means is that system availability would increase if these steps are taken.

Software Maintenance Needs

The software maintenance needs are no different in that system availability would still be prime on the user's mind. When the user requires assistance, he would like to have the facility to call through the telephone to obtain quick resolution to his problem. However, if the problem cannot be resolved though the telephone, on-site assistance would be required.

Temporary work-arounds would be required if no resolution is forthcoming. In the situation when problems are "grey", assistance should be made available to aid the user in determining the cause of the problem. Other forms of assistance include providing emergency patches and escalating the problem for the vendor to muster more resources.

Software installation activities, software and manual updates are some of the other software maintenance needs which a user may require.

There are essentially five main criteria for choosing a computer maintenance service. They are:

1 Criticality of the application systems,
2 Multiple systems and sites,
3 Security and geographic considerations,
4 Level of in-house expertise and
5 Price/Performance.

1 Criticality of the Application Systems

The criticality of the application systems is the primary determinant for the hours of coverage and response time required for your computer maintenance service. If the production environment does not permit a prolonged period of down-time, then the hours of coverage required would have to be extended. The longer the coverage hours, the higher the cost.

In the same vein, the response time would also be a point of consideration in order to ensure that the system availability is maintained. The shorter the expected response time is, the higher the cost. This should of course result in higher system availability.

2 Multiple Systems and Sites

Sites with multiple systems would require different level of system maintenance service compared to those with single computers. In a single site, multiple system environment, the configuration in the computer room can be arranged to ensure that some peripherals can be shared to increase the system availability. If such arrangement is possible, the criticality of having shorter response time and longer hours of coverage is reduced.

In the situation where multiple systems are located at different sites, the level of system maintenance depends on whether the system management of such operations is centralized or decentralized. This determines the type and level of service expected from the vendors.

Generally, operations which choose to manage the systems in a centralized fashion are those with centralized application development and technical support teams. The remote sites are usually run-only environments with minimal operator intervention.

A centralized system management strategy can affect the way manual and software updates are co-ordinated. Similarly, telephone assistance can be co-ordinated from one technical focal point. It also determines how the application and system software are distributed. Such an environment usually requires system compatibility across the various operations. This can be ensured with proper co-ordination between multiple sites. It is desirable in this environment to have software maintenance co-ordinated at the central site.

The hours of coverage and response times required for hardware maintenance at different sites will depend on the needs. However, the maintenance planning activities can still be co-ordinated from the central site.

3 Security and Geographical Considerations

In some operations, the consideration could be one of security and/or geography. This point of consideration usually leads you to examine the viability of self-maintenance. Some of the activities which users can undertake include work-station maintenance, first line system diagnostic, memory dump analysis, board-level changes, system performance tuning, etc.

Users may want systems to be self-maintained instead of depending on vendors for security reasons. Security requirements may make it impossible to allow engineers to troubleshoot freely and expeditiously. Similarly, if there are sites which are remotely located away from the nearest vendor service center, then it may necessary to think about some form of self-sufficiency so that the system availability will not be affected by the long time it takes for an engineer to be on-site. The user, however, would have to think about the burden of attending specialist courses on an ongoing basis. On top of that, the user is also expected to carry the cost of spare parts inventory. There is the consideration about product obsolescence and users may be required to carry this burden too.

4 Level of in-house expertise

The degree of in-house expertise of the MIS staff and the size of operation are also points for consideration in determining the level of maintenance service. Generally, a higher technically competent team supporting a large operation requires a different set of features in the service compared to a less knowledgeable user. They would have a more sophisticated set of requirements like advanced education courses, in-house performance tuning capability, first-line hardware and system software troubleshooting, etc. They too would have the critical mass in work and personnel to justify for this level of service than a smaller operation.

5 Price/Performance

In the final analysis, the cost factor would be heavily featured. This determines the price-performance of the maintenance service. The above discussion shows that the higher the level of service, the higher the costs. However, if cost is a concern, then the user must be prepared to accept a lower level of system availability and set of services from the maintenance program.

In HP, it is recommended that users take a Standard Systems Maintenance Service if he has a application critical environment or Basic Systems Maintenance Service for cost sensitive applications and when some system down-time can be tolerated.

Software support offered by HP varies in the level of service and cost. For very experienced users who may be in a static and/or run-only environment, it is recommended that they opt for a Response Center Support service. In most cases where users are still in development and may have networked systems, Account Management Support service is recommended.

FINANCIAL ANALYSIS FOR COMPUTER SYSTEM MAINTENANCE

Now that we have covered the features and examined the cost factors involved in the planning for maintenance, the next key issue of the exercise is the justification of the investment.

The methodology to justify the maintenance service from the cost point of view is indeed a very simple one. It involves costing the MIS operation and determining how much it would affect the operation if the system is down.

Assuming that it costs US$190,000 to run a small MIS operation. The US$190,000 can be broken down into three main components - salary for system manager/analyst, programmer/operator accounts for US$60,000; operational overheads takes up US$120,000 and consumables US$10,000. Operational overheads typically comprise communications, occupancy, training, expense equipment, hardware maintenance, etc. Assuming that there are 250 working days in a year, this is translated to be US$760 per day. In other words, it would cost $760 per day to run an MIS operation. A typical Account Management Support contract for a Micro 3000/XE system from HP costs about US$5,000 per annum or translated into MIS operational costs, it is equivalent to 7 (5000/760) production days.

Translating 7 days into actual work done at the MIS department, this value can be quite significant. Viewed from a different perspective, for an additional US$5000 annual cost incurred in software support, the MIS department may gain up to 7 days worth of production time. There are other associated benefits which the software maintenance program offers but cannot be easily quantified here. The figures would be more if the end-user operations are taken into consideration, and of course this is dependent on the kind of business the company is in. The figures would be easier to justify if the end-user impact is taken into consideration. These will far exceed the cost of the software contract.

The above argument also holds for hardware maintenance. We could put in the numbers for both hardware and software maintenance and go through the same exercise. It would arrive at the equivalent production days. This would form the basis for your justification.

TRENDS IN THE COMPUTER SYSTEM MAINTENANCE

Trends in the computer industry have changed over the years from one of reactive in nature to being more pro-active and preventive. There is a very noticeable trend to move towards remote electronic support. Problems are beginning to be solved more through the public telecommunications network. There frequently is now lesser dependence on on-site visits by the engineers. With prior diagnosis through remote electronic support, the engineers, during on-site visit, will be better prepared with the right tools and spare parts thus shortening the problem resolution time. Besides, there is a larger source of information to help in the troubleshooting process if the engineer analyzes and researches the problem from the office.

Temporary work-arounds can then be implemented quicker because there is a clearer understanding of the problem at the engineer's office. Not only would users be able to receive temporary solutions faster, the service cost is also expected to drop too.

There is also a trend of centralizing information for access by both customers and engineers alike. Such kind of service is beginning to be introduced in the computer industry by hardware vendors. This service enables the MIS staff to provide first-line troubleshooting by researching into a large centralized knowledge database. This is going to be the key to help bring down the cost of servicing users and at the same time, increase the technical competence of the technical people in the user base.

Predictive support capabilities of computer hardware are proliferating in many areas. This capability allows built-in software to predict potential failures and alarms the engineers and users on the impending problems. Action steps can then be taken to avert potential down-time. Disruption to the system can thus be minimized and planned. This helps to increase user productivity. With this mode of operation, active steps can be planned and manpower needs lowered, thus lowering service costs.

The advances in artificial intelligence have made it possible to think about the possibility of applying it in the field of service maintenance. Memory dump analysis is one area of application. Built-in diagnostic is another area which can be incorporated in the system hardware.

CONCLUSION

There is always a cost and performance trade-off. Usually the higher and more urgent the need, the higher would be the cost of the maintenance program.

It is not difficult to select the right level of computer system maintenance service if you can systematically list down the maintenance needs of your operation. A list of selection criteria can then be designed to meet the organization's maintenance needs. Cost and benefit analysis is then conducted to justify your selection of a particular maintenance program.

COST JUSTIFYING OFFICE SYSTEMS

by
Bill Franklin
Hewlett-Packard
19091 Pruneridge Avenue
Cupertino, California, 95014

CHALLENGES:

It has often been said that Office Automation is

* Impossible to cost-justify
* Can not provide measurable results
* Must be accepted by senior management in a "leap of faith" if it feels good.

While many managers believe that it makes business-sense, they can not effectively convince top management that the company will achieve an appropriate return on its office automation investment.

In many organizations, investments in office automation are evaluated against other alternative profit producing opportunities. Since many organizations have limited financial resources, the challenge is to determine where they can get the greatest return on their investment. This makes business sense. Companies are in business to make money and sound business decisions are based upon a careful analysis of "expected gains" from alternative "application of funds".

Supervisors in factories can count the products "rolling off" the assembly line and determine the benefits of computer automation efforts. However, in offices, efficiencies are not as easily tallied.

Top management is evaluated by profits and earnings. According to an April 28th 1986 Fortune article, CEOs cited their most important objectives as "Improved profits, earnings" and "growth". The issue of "cost containment" was sited by 78% of the CEOs and "productivity" by 77%. Almost 40% of the CEOs responding indicated that progress was slow or little has been made. Office Automation can provide a real value to organizations in helping them realize these objectives.

I am convinced that Office Automation can be cost justified and is sound business investment for many organizations.

In this paper techniques will be presented to help managers cost justify office automation.

EXAMPLES OF APPLYING TECHNOLOGY:

The following are 3 examples of applying technology to solve problems:

COMPUTATION OF PI: Yasumasa Kaneda, a Tokyo assistant professor, calculated pi to 201,326,000 digits. Since most practical applications require a calculation of 10 digits and extremely exact applications only need as many as 30 places, the questions that must be asked are: Why? What value does this have?

ELECTRONIC MAIL TAUGHT CLASS: Professor Gerald Phillips of Penn State University communicates with his Speech Communications class only through the schools Electronic-mail. The class gets weekly lectures from graduate students and other graduate students sit in on group discussions as consultants. Phillips doesn't come to class. Last semester he answered 3,169 questions from his approximately 200 students. This, in theory, could give the students access to the best minds in the country for classes.

AUTOMATIC RADAR DEVICE: The city of Paradise Valley in Arizona has contracted with a firm to help them catch speeders. The firm has technology that checks the speed of vehicles moving faster than a designated speed and takes a picture of speeding vehicles. Owners are identified by their license and the firm mails citations to the owners. The device is capable of issuing 260 citations per hour compared to the police's 2 to 4 per hour. Up to 100 speeders a day are caught. The manufacturer loans the city the device, but for every ticket that the firm writes, it keeps $20. This is a WIN-WIN situation. The city's bottom-line revenue picture is definitely improved and even with the equipment costing $42,500, the firm would break-even within 22 days.

The goal of office automation is to implement technology where there is a quantifiable benefit - a WIN-WIN situation. In many cases the return is more important than the amount invested.

STAGNATION:

There has been some "stagnation" in the improvement of white-collar productivity. In order to achieve increased results management should:

Measure the "return on management".
A plant manager may be rewarded for getting more work from

fewer workers. However, a business professional (manager) is many times rewarded based upon the size of his organization. As CEOs seek to determine the payoff from the staffs they have built, there will be more incentive to justify these staffs. Rewards could be more closely tied to performance than the size of the staff required to do the work. This will increase the focus on office automation.

Increase the scope of automation to business professionals. Historically there has been much effort to automate the secretarial staff and cost justify the investment through labor savings that are planned to lead to staff reductions. However, this method of analysis rarely produces the expected results.

Encourage training and the effective use of new technology. Without changing the way employees perform tasks, all the technology in the world will not increase productivity.

HISTORY:

In the 1970's hardware was so expensive that systems were viewed as "cheep" if there was excess capacity on an expensive mainframe-processor. It made economic sense to keep the expensive-processor busy. An investment in software and people could be easily cost-justified. Organizations are now faced with the trade-offs of "One person for one year or a one MIP (millions of instructions per second) processor". With the hardware component of office dropping and the increase use of personal computers, office software systems can be more easily justified.

Many data processing projects were justified by quantifying the tangible and intangible benefits that would be accrued to the project. Many of these efforts historically focused upon the displacement of people. These forecasted displacements did not always materialized. Many company's personal policies and unions do not look favorably upon this type of activity.

There has been much discussion of a "paperless office". It was expected to produce major cost-savings in reducing the amount of paper that was used. While some automation products have the capability to print four pages or more on a single page, the goal of a "paperless office" has not materialized. Through the application of software some paper is saved. However many "office workers" feel the need for "hard-copies" and do not use "electronic-filing technologies". When "office workers" change the way they work, paper in the office will be further reduced.

POTENTIAL MARKET:

According to the U.S. Department of Labor "office workers" comprise 39.8% of the U.S. workforce, or 40.7 million employees. Salaries of the managers, professionals, secretaries, and administrators amount to over $900 billion annually. The 800,000 users of integrated office systems comprise only 2% of the total possible number of end-users.

Market potential is impacted by management's understanding of how benefits are derived.

The following examples demonstrate how the office automation potential expands with effective implementations:

1. Networking and information sharing can increase the benefits. A stand-alone PC has far less potential value than one linked into a network that allows for information sharing and electronic-mail.

2. Integrated systems maximize office automation use. Easy-to-use systems stimulate acceptance.

3. Office automation is not a substitute for effective planning. An excellent system can be designed around an ineffective process. The process can be made to work easier, but it is still not effective.

TOP MANAGEMENT NEEDS:

In demonstrating value of a project to top management, many times cost justifications are required. There are only two ways to impact the bottom-line: increase sales or reduce costs.

There is some information that is very important to have prior to developing a cost justification:

1. <u>Hurdle Rate.</u> How much needs to be justified before acceptance?

2. <u>Project Selection Method.</u> How does top management evaluate projects? (IRR, NPV, Payback., etc.) It is important to know the factors that they will be using to select a project in which to invest.

METHODS:

There have been some very elaborate processes developed to cost justifying office automation investments. Some of these focus around the following:

* Mean time saved by technology times the salary of those saving time.
* Time saved times pre-tax income resulting in opportunity savings.
* Computing savings but not determining the bottom-line effects on the organization.
* Focus on the professionals in the organization and determine their value to the company (the value of management + value of business systems divided by the cost of management + cost of the business systems). It is assumed that organizations can get greater benefit from automation of those who have a greater impact on organizational profits.

The processes that will be explained in more detail in this paper are:

1. <u>HEDONISTIC WAGE MODEL.</u> Determine the amount of "lower level" work being performed by "higher level" workers and derive the impact on reducing the improper allocation by automation.

2. <u>ACCOUNTING ORIENTED MODEL.</u> Evaluate potential contribution to profits of automation focusing on those that are measurable and quantifiable.

3. <u>PROCESS-ORIENTED MODEL.</u> Evaluate processes and determining the potential cost savings and revenue increases that are expected to accrue from automation.

HEDONISTIC WAGE MODEL:

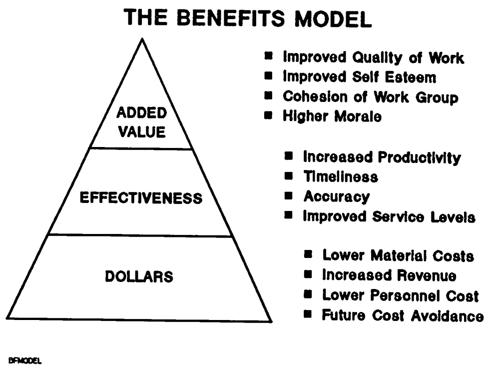
The "Hedonistic Wage Model" was referenced in a study by Sassone and Schwartz in a February 1986 issue of "Datamation".

This model assumes that workers are worth what they cost the company in wages, salaries, benefits, and overhead. Their worth can be thought of as the weighted values of the activities that they perform. For example, assume that secretaries spend 85% doing work and are unproductive 15% of the time. In this case the cost of clerical work would be their salary divided by .85. If business professionals spend 50% of their time doing professional work, 35% doing clerical work, and 15% unproductive, then the value of their work would be a function of the portion of time doing professional work as well as clerical and unproductive work which is computed at a lower value.

The goal of office automation is to increase the amount of time that can be spent on higher-level tasks.

ACCOUNTING ORIENTED MODEL:

This "accounting oriented" model breaks benefits down into 3 categories:



1. HARD DOLLAR BENEFITS - Measurable and quantifiable - direct cause and effect:

These are obtained where there is a 100% direct cause and effect relationship between an expenditure in technology and a resulting impact on the bottom line.

Examples of this type of benefit are: lowering material costs; increasing revenue; lowering personnel costs; future cost avoidance; reducing material costs; reducing the use of outside services (courier services, service bureaus, temporary help); eliminating existing, filled positions; selling unneeded assets (displaced workstations, copiers, file cabinets); measuring a direct improvement in cash flow because of Automated Office Systems (more timely billings and collections); space savings if more space is needed; and avoiding costs if planned (budgeted positions disappear).

2. <u>IMPROVED EFFECTIVENESS</u> - Measurable and quantifiable but not auditable (productivity - getting sales proposals to the customer faster):

This type of benefit is obtained, where there are potentially multiple contributors to an overall change, not just an investment in office automation.

Examples of this type of benefit are increased productivity, timeliness, accuracy or improved service levels; improving the service rates, if this impacts revenue; and quantifying increases in revenues through automation efforts.

3. <u>ADDED VALUE</u> - Perceivable and measurable but not auditable (improvements in self esteem, cohesion, etc.):

This type of benefit is usually associated with the user of the technology, or the people impacted by the technology. An example of this is customer's or supplier's perception of the changes brought about by the technology.

The benefits which can be obtained in this category include improved quality of work, improved self esteem, increased cohesion of work group or higher morale; improving the quality of the employees worklife; better and more timely information for decisions; improved employee morale; reduced error rates; and savings un-budgeted dollars.

These benefits are usually more intangible.

There is a tendency to ignore "intangible benefits" because it is said that they can not be quantified. Ignoring them assigns a value of zero (0) to them in determining ROI, NPV,and Payback. Intangibles can be quantified by using the technique outlined below:

Break down an intangible benefit in its component parts. For example, office automation could "Improve the quality of the employees worklife" through the reduction of redundant / boring tasks. This could result in lower turnover, lower absenteeism, faster promotions, faster learning of new jobs, and higher productivity. Lower turnover reduces recruiting, replacement, and training costs as well as reduces the extra supervision that is often required during the learning period. These costs are easier to quantify (create time and dollar estimates) than the intangible benefit. Personnel departments have a record of some of these costs. Recruiting costs can be a major expense. A study by a California recruiting firm noted that exempt hiring costs (including advertising, travel, agency fees, relocation costs, interviewing, etc.) run between $3,000 to $27,000 per person with an average of $15,000 while non-exempt employee recruiting costs range from between $600 to $6,000 with an average of $3,300 per person. These do vary by industry and by job classification and the data is available from personnel departments and recruiting firms.

PROCESS-ORIENTED MODEL:

In order to understand the potential impact of office automation on an organization, it is important that a company clearly understand the inner workings of their organization down to the workgroup and individual level. By understanding the current organization, a baseline is established and the type of work being done is understood. In addition, areas of potential improvement become obvious.

The following steps can be helpful in developing process-oriented cost justifications:

1. Determine the area of focus / business problem or need to be addressed. Is there a deviation from industry / activity norms.
2. Understand the current process. Diagram the process for understanding and analysis.
3. Determine how automation can address the business problem or need. What is quantifiable? What is positive impact that office automation makes?
4. Develop the cost justification focusing on "cost-displacement" and "value-added".

In determining how automation can address the business problem or need, the following should be considered:

<u>ADDRESSING THE NEED / IMPROVING THE PROCESS</u> - How does office automation technology make a positive impact on the organization?

* Do the tasks take less time?
* Are tasks completed in a more timely fashion?
* Are tasks not previously being being done now?
* Are tasks done with greater quality / success?
* Is there greater job satisfaction?

<u>DETERMINE THE BENEFIT</u> - Why is change good for the organization?

* Tasks take less time: Where does the time go?
* Tasks are completed in a more timely fashion: What is the rush?
* Tasks not previously being done: Why are they worth doing?

* Tasks done with greater quality / success: Who appreciates quality?
* There is greater job satisfaction: Why pay for satisfaction?

QUANTIFY THE BENEFITS - What is the value of the "good"?

* Increased revenues can occur when the tools are used to generate sales.
* Can increase the odds of closing some sales (quantifiable gain).
* Can reduce the risk of a quantifiable loss.
* Reduce operating and production costs.
* Accelerate revenues.

PROCESS COST JUSTIFICATIONS:

While each organization in unique in how they address problems, there are some "common processes" that can are applicable to other organizations. This section contains a few worksheets that can be used to demonstrate the value of office automation to management.

PROCESS:

In order to perform some tasks there are steps that must be followed.

1. Use the "Need Determination Questions" to help establish top management awareness of the need and user support to helping justify the solution.
   a. HIGH LEVEL QUESTIONS:
   Used when working with managers who are measured by profits, evaluated by profits, and accountable for profits. It is important to verify that the need is understood and supported by high level management before spending an excessive amount of time at the user level.
   b. USER LEVEL QUESTIONS:
   Used when working with those who are involved in doing the work. In order to tailor assumptions to your organization, it is important that the data be verified with those whose needs the solution is addressing.

2. Assume that management and the user understands the process. Use graphics to help explain the process.

3. Use the baseline metrics (in parentheses) as a starting point for the justification. By refining the baseline

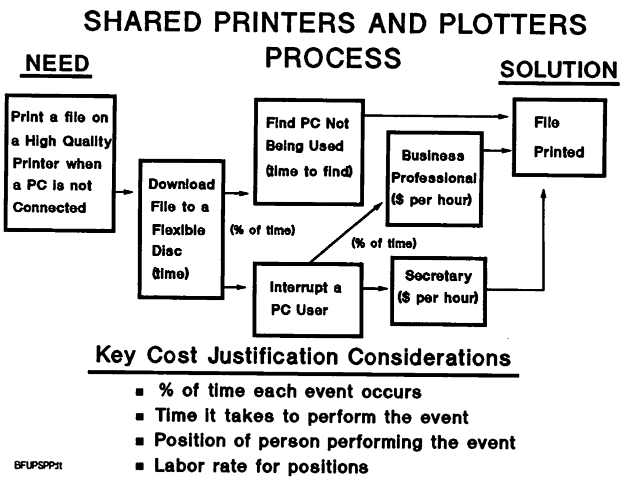
metrics to reflect your organization's situation, individual cost savings and revenue increases can be computed.

4. Verify numbers with knowledgeable people within your organization. Be sure to verify that the assumptions you are using are correct.

5. Compile this data into a cost-justification model.

SHARED PRINTERS AND PLOTTERS

An example of this is a common business scenario of printing a file on a high quality printer when the user does not have one attached to his/her personal computer. There are a finite number of options as can be seen from the following diagram.



Shared printers and plotters allow users to access high quality printers from their PCs. This feature saves the costs of the alternatives (forgo high quality documents which might translate into reduced sales or force users to search out the designated printers located in the building). This section focuses on the time lost looking for a common printer and interrupting another person working there.

NOTE: The potential sales gains from printing on high quality printers is at the end of this section.

I. NEED DETERMINATION QUESTIONS:

A. <u>High level</u>:

1. Do professional looking documents have an affect on revenues?
2. How much time do you believe is "wasted" by business professional trying to get documents printed?
3. Are you missing any important due dates on the delivery of proposals or other material that impacts potential revenues?
4. How important is quality printing to your organization?

B. <u>User Level:</u>

1. How do you currently get your documents printed?
2. Do you find that you "waste time" trying to get documents printed?
3. How much more productive would you be if you had direct access to high quality printers?
4. Would you print additional documents on high quality printers if it were easier to use?

II. WORKSHEET:

A. Savings from Shared Printing

<u>Situation #1: Professional User who searches for printers:</u>

Average time to copy a file (document / spreadsheet / graph) to be printed to a diskette: (1 minute) . . . . _________

Average time to reload file to be printed: (1 minute) . . . . +_________

Average time wasted per file looking for a PC connected to a printer: (3 minutes) . . . . +_________

Average time spent walking to a PC connected to a printer: (2 minutes) . . . . +_________

Number of files that are printed on high quality printers per month per PC not connected to a high quality printer: (40) . . . . X_________

% of time user of a terminal / PC is not interrupted (75%) . . . . X________

Labor rate for Business Professional ($65 per hour / 60) . . . . X________

TOTAL for Situation #1 ($227) . . . . =________

<u>Situation #2: Professional User who interrupts someone connected to a printer:</u>

Average time to copy a file (document / spreadsheet / graph) to be printed to a diskette: (1 minute) ________

Average time to reload file to be printed: (1 minute) . . . . +________

Average time spent walking to a PC connected to a printer: (2 minutes) . . . . +________

Number of files that are printed on high quality printers per month per PC not connected to a high quality printer: (40) . . . . X________

Labor rate for Business Professional ($65 per hour / 60) . . . . X________

TOTAL for Situation #2 ($173) . . . . =________

B. Savings by the person being interrupted:

<u>Situation #1: Interrupt Business Professional:</u>

Number of files that are printed on high quality printers per month per PC not connected to a high quality printer: (40) . . . . ________

% of time user of a terminal / PC is interrupted (25%) . . . . X________

% of time user is a business professional (30%) . . X________

Average time spent stopping tasks, waiting for file to be printed and restarting tasks by person interrupted: (3 min.) . . . . X________

Labor rate for Business Professional ($65 per hour / 60) . . . . X________

TOTAL for Situation #1 ($10) . . . . =________

<u>Situation #2: Interrupt Secretary:</u>

Number of files that are printed on high quality printers per month per PC not connected to a high quality printer: (40) . . . . X________

% of time user of a terminal / PC is interrupted (25%) . . . . X________

% of time user is a secretary is interrupted (70%). X________

Average time spent stopping tasks, waiting for
file to be printed and restarting tasks
by person interrupted: (3 min.) . . . . X________
Labor rate for Secretaries: ($35 per hour / 60) . . X________
TOTAL for Situation #2 ($12) . . . . =________

C. **SUMMARY OF COST SAVINGS (A - B ABOVE):**
1. Professional User who does not interrupt others ($227) . . . . ________
2. Professional User who interrupts others ($173). . . . . +________
3. Interrupted person - Business Professional ($10). . . . . +________
4. Interrupted person - Secretary ($12). . . . . +________
5. SUBTOTAL ($422) . . . . =________
6. Number of PCs connected to system (20). . . X________
7. Number of months per year (12) . . . . X________
8. TOTAL SAVINGS ($101,280). . . . . =________

D. **Potential Sales gains as a result of professional documents being sent to the customer:**

Sales ($500,000). ________
Potential sales gains from professional looking
documents (% of sales): (.05%) . . . . X________
Profit Margin on sales (% of Sales) (10%) . . . . . X________
TOTAL potential sales gains ($2,500). . . . . =________

DATA BACKUP:

Data backup provides reliable and secure backup on PC hard disc data. To assess the value of this backup, estimate the probability and cost of losing PC data. The following sections detail the costs in recreating text documents, graphs and spreadsheets. There is also a place for valuing PC applications like Lotus 1-2-3 which would need to be repurchased in situations where a hard disc is completely erased. It is assumed that:

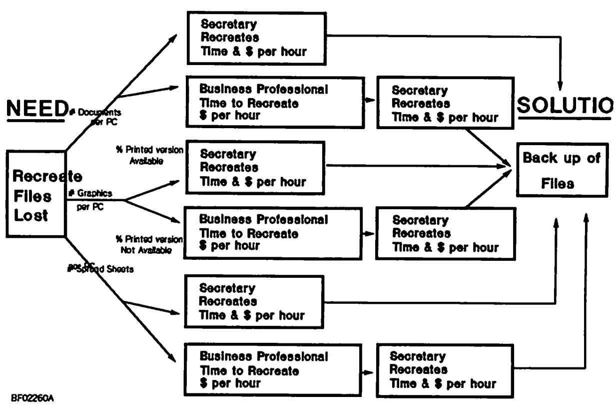
1. Secretaries have more graphs on their systems than spreadsheets - figures will need to be adjusted based upon your organization's mix.

2. Drawings do not change as frequently as documents and therefore a copy is more likely to be the current one.

3. In recovering files (documents, graphs, drawings and spreadsheets) it is assumed that the secretary will perform as much data entry as possible while the business professional will evaluate and perform the recreation of the content.

4. There will be various degrees of recreation required.

DATA BACK UP PROCESS



I. NEED DETERMINATION QUESTIONS:

A. High Level:
1. How much is the data / information worth to you?
2. Will any decisions be affected by not having it when needed?

B. User Level:
1. What would you do if you lost all the data on your hard disc?
2. Could you recreate all the data?
3. Are you sure that you could recover it correctly?

II. WORKSHEET:

A. Savings in Dollars for Recreating Documents (per PC):

Situation #1: Printed version not available:

Average Number of Documents on 20MB Hard Disc: (50) . . . . ________
% of them that would need to be 100% recovered: (50%) . . . . .X ________
Average number of lines per document: (300) . . .X ________
Time per line to re-enter from printed version: (30 sec. / line.) . . . . .X ________
Labor rate for Secretaries: ($35 per hour / 3600) .X ________
Subtotal Cost for Secretary ($2,187) . . . .= ________

Average Number of Documents on 20MB Hard Disc: (50) ________
% of them that would need to be 100% recovered: (50%) . . . . X________
Business Professional time required to recreate document content: (3 hours per document) . . . X________
Labor rate for Business Professional ($65 per hour) X________
Subtotal Cost for Business Professional ($4,875) . =________

Subtotal Cost for Secretary ($2,187) . . . . ________
Subtotal Cost for Business Professional ($4,875) . +________
Total Dollar Savings for those documents where printed version is not available ($7,062) . . . =________
% of documents where a current copy is not available: (10%) . . . . X________
TOTAL for Situation #1 ($706) . . . . =________

Situation #2: Printed version available:

Average Number of Documents on 20MB Hard Disc:
(50) . . . . ________
% of them that would need to be 100% recovered:
(50%). . . . . X________
Average number of lines per document: (300) . . . . X________
Time per line to re-enter from printed version:
(30 sec. / line.). . . . . X________
Labor rate for Secretaries: ($35 per hour / 3600) . X________
% of documents where a current copy is available:
(90%) . . . . X________
TOTAL for Situation #2 ($1,969) . . . . =________

B. Savings in Dollars for Recreating Graphs / Drawings (per PC):

Situation #1: Printed version not available:

Number of Graphs or Drawings on 20MB Hard Disc
(10). . . . . ________
% of them that would need to be 100% recovered:
(50%) . . . . X________
Time per graph or drawing to recreate from
a printed version: (1 hour). . . . . X________
Labor rate for Secretaries: ($35 per hour). . . . . . X________
Subtotal Cost for Secretary ($175) . . . . =________

Number of Graphs or Drawings on 20MB Hard Disc
(10). . . . . ________
% of them that would need to be 100% recovered:
(50%) . . . . X________
Business Professional time required to recreate
a graph and/or drawing content: (30 min. per
graph or drawing) . . . . X________
Labor rate for Business Professional
($65 per hour / 60) . . . . X________
Subtotal Cost for Business Professional ($163) . . =________

Subtotal Cost for Secretary ($175) . . . . ________
Subtotal Cost for Business Professional ($163) . . +________
Total Dollar Savings for those graphs / drawings
where a printed version is not available ($338). =________
% of graphs and drawings where a current copy is not
available: (5%) . . . . X________
TOTAL for Situation #1 ($17) . . . . =________

Situation #2: Printed version available:
Number of Graphs or Drawings on 20MB Hard Disc (10) . . . . ______
% of them that would need to be 100% recovered: (50%) . . . . X______
Time per graph or drawing to recreate from a printed version: (1 hour) . . . . X______
Labor rate for Secretaries: ($35 per hour) . . . . X______
% of graphs and drawings where a current copy is available: (95%) . . . . X______
TOTAL for Situation #2 ($166) . . . . =______

C. Savings in Dollars for Recreating Spreadsheets (per PC):

Situation #1: Printed Version not Available:
Number of Spreadsheets on 20MB Hard Disc (5) . . . . ______
% of them that would need to be 100% recovered: (50%) . . . . X______
Time per spreadsheet to recreate from a printed version: (30 min.) . . . . X______
Labor rate for Secretaries: ($35 per hour) . . . . X______
Subtotal Cost for Secretary ($44) . . . . =______

Number of Spreadsheets on 20MB Hard Disc (5) . . . . ______
% of them that would need to be 100% recovered: (50%) . . . . X______
Business Professional time required to recreate spreadsheet content: (3 hours per spreadsheet) . X______
Labor rate for Business Professional ($65 per hour / 60) . . . . X______
Subtotal Cost for Business Professional ($488) . =______
Subtotal Cost for Secretary ($44) . . . . ______

Subtotal Cost for Business Professional ($488) . . +______
Total Dollar Savings for those graphs / drawings where a printed version is not available ($532) . ______
% of spreadsheets where a current copy is not available: (10%) . . . . X______
TOTAL for Situation #1 ($53) . . . . =______

Situation #2: Printed version available:
Number of Spreadsheets on 20MB Hard Disc (5). . . . ________
% of them that would need to be 100% recovered:
(50%) . . . . X________
Time per spreadsheet to recreate from a printed
version: (30 min.) . . . . X________
Labor rate for Secretaries: ($35 per hour / 60) . . X________
% of spreadsheets where a current copy is
available: (90%) . . . . X________
TOTAL for Situation #2 ($39) . . . . =________

D. Cost of Software to Recover per PC:

Average cost per copy to get replacements: ($500) . ________
Average number of copies per PC: (3) . . . . X________
TOTAL for Cost of Software to Recover per PC
($1,500) . . . . =________

E. SUMMARY OF COST SAVINGS (A - D PREVIOUS PAGE)

1. Labor Savings for recovering documents:
   a. Printed version not available ($706). . ________
   b. Printed version available ($1,969). . . +________
2. Labor Savings for recovering graphs / drawings:
   a. Printed version not available ($17) . . +________
   b. Printed version available ($166) . . +________
3. Labor Savings for recovering spreadsheets:
   a. Printed version not available ($53) . . +________
   b. Printed version available ($39) . . . . +________
4. Cost of software to recover ($1500) . . . . +________
5. SUBTOTAL ($4,450) . . . . =________
6. % probability of a disc crash and loosing data (5%) . . . . X________
7. Number of PCs connected to system (20). . . X________
8. TOTAL SAVINGS ($4,450). . . . . =________

F. Average estimated contributions to profits as a result of timely information for decisions per PC per month from:

NOTE: Every document, graph, drawing, spreadsheet has value and should contribute to profits - otherwise, why was it created?

Documents: ($100) . . . . ________
Graphs / Drawings: ($100) . . . . +________
Spreadsheets: ($100). . . . . +________
TOTAL for Contributions to Profits ($300) . . . . . =________

ELECTRONIC MAIL:

Some changes involve standardization. While there is a cost to individual freedoms from standardization, there are also benefits that accrue to the organization. These need to be weighted and understood. If everybody is expected to use electronic-mail then this constraint will cause a change in behavior. However, it will have greater value. Imagine the total value to installing an electronic-mail system where only one one person ever uses the system. Sending messages to yourself has little value. The real value is a result of a total workgroup or organization using the system. If you send a message to someone and they never read their mail then your message is lost.

HP has experience in using electronic mail networks. It is estimated that HP's electronic mail network is between the 7th and 8th largest in the U.S. in terms of registered users.

The use of HPDeskManager has increased to such an extent that it is now the communication backbone for Hewlett Packard with over 65,000 registered users on the network. It costs approximately $58 per month per registered user.

Our average delivery time anywhere in the world is 7 hours using "normal service" and one hour using "urgent service". What value does this have to an organization? The time value of information needs to be considered.

The average cost per page is $.61. The average number of recipients per message is 2.075. The cost per recipient is $.29.

The Interactive Office Systems Group, which is responsible for the internal implementation of HPDeskManager calculated the costs of sending information between HP offices using a variety of different media. The costs which were included in this analysis covered depreciation, support, service, operations, facilities etc. to obtain a total cost for the service.

The comparative "dollar costs" for sending two pages of information per recipient using a variety of media are:

System	2 Pages	Rate
HP Interoffice Mail	$0.10	$0.30 per ounce
US Postal Service	$0.44	$0.44 per ounce
HP's HPDesk Network	$0.58	$0.29 per 4000 bytes
Public E-Mail Net 1	$1.60	$0.80 each 7500 bytes
Public E-Mail Net 2	$2.00	$1.00 each 7500 bytes
Fax	$3.20	$2.39 1st page, 0.81 each additional page
Express Mail	$10.00	1 - 30 pages for 10.00

The only mechanisms which are lower in cost for distributing information than HPDeskManager were "HP's interoffice mail" and the "US postal service". However they are much slower mechanisms for distributing information than HPDeskManager. Information has a time value to it. When document preparation time, photocopying time, and paper costs are included, these services are more cost comparable to HPDeskManager.

HPDeskManager was introduced to the U.S. sales force last year. Remote access to the system was made possible through the use of portable computers.

Richard S. Burgess, a HP marketing manager, was quoted in the Oct. 12, 1987 issue of Business Week on "HP's sales force automation project". He indicated, "We changed the way people communicate with their boss and peers". As a result of HP's project, time with customer's increased 27% and sales rose by 10% in the pilot group. Even if sales only rise by half as much, the 1700 reps that will become automated this year, will generate $30 million in pretax profits. This will be 5 times the project cost.

Other benefits from this project have been: greater sales-force enthusiasm, confidence and professionalism; increased creativity in performing their jobs; and a more team-minded spirit.

One company that sells a wide range of construction and home maintenance products installed electronic-mail services between its field sales staff and its headquarters. The result was instead of returning unsold inventory to the company, distributors around the country could communicate and ship products directly to areas that needed them. Swapping of returned goods increased by 70%, preventing lost profits of $70,000 per year. One-year ROI was 530% and 2,300% over five years.

For electronic mail, the following analysis focuses on basic benefits: reducing phone tag, verifying whether messages were received and sending messages. This analysis assumes that users want messages as quickly as possible, and this translates into business benefits.

NOTE: While there are some benefits to electronic mail, the objective is to present a simplified approach on how electronic mail could be justified. There are many features of HPDeskManager that could be included in the analysis depending upon the application and industry.

I. NEED DETERMINATION QUESTIONS:

A. High Level:

1. Do your employees spend a significant amount of time playing "phone tag"? (an unproductive activity)
2. Are any decisions affected by not getting a simple answer in a timely fashion?
3. Is it important to you to have current information easily transferred between people within your organization?

B. User Level:

1. How many times do you call someone only to leave a phone message?
2. How many times do you need to assure that a message is sent and read by someone in a remote location?

II. WORKSHEET (from metrics above):

A. Labor savings in communication - phone tag:

Situation #1 - Simple response desired:

Number of pink slips generated per month by a PC user calling someone who was not at their desk (100) . . . . ________

% of calls where a "pink slip" was generated when a simple response could have been created using electronic mail (20%) . . . . X________

Time it takes per call to create a pink slip: (2 minutes) . . . . X________

Labor rate for Business Professional: ($65 per hour) . . . . ________

Labor rate for Secretaries: ($35 per hour) . . . .+________

Total combined labor to give and take message ($100 per hour / 60) . . =________ -> X________

Total labor savings in phone tag per PC / terminal user ($66). . . . . =________

Situation #2 - Verifying message received or read:

Number of pink slips generated per month by a PC user calling someone who was not available (100). . . . . ________

% of calls where a "pink slip" was generated where the purpose was to verify that the message was read or received (10%) . . . . X________

Time it takes per call to create a pink slip: (2 minutes) . . . . X________

Labor rate for Business Professional: ($65 per hour) . . . . ________

Labor rate for Secretaries: ($35 per hour) . . . .+________

Total combined labor to give and take message ($100 per hour / 60) . . .=________ -> X________

Total labor savings in phone tag per PC / terminal user ($33). . . . . =________

B. Labor savings in communication - sending messages:

Creating labels, stuffing envelopes, etc. (1 minute).________

Number of messages (newsletters, standard reports, expense reports, sales figures) that need to be sent per person: (100) . . . . X________

Average number of people who need to be copied per message: (5) . . . . X________

Secretary time per message to copy it for distribution (going to copier, waiting for someone else to finish copying, actual copy time): (2 minutes) . . . . X________

Labor rate for Secretaries: ($35 per hour / 60) . . X________

Total cost of sending messages per month per PC user ($875) . . . . =________

C. Material costs in sending messages by mail:

Number of messages that need to be sent per month per person: (100) . . . . ________

Average number of people who need to be copied per message: (5) . . . . X________

Cost of mailing correspondence using the post office (including envelope, stationary, stamps, mail room costs, etc.): ($1.00 per letter). . . . . X________

Total material cost per PC user ($500). . . . . . . =________

D. Additional savings in communication - sending messages that need to be sent overnight:

Additional Labor Savings:
Number of messages that need to be sent per month per person: (100) . . . . ________
Average number of people who need to be copied per message: (5) . . . . X________
% of messages that need to be sent overnight (5%) . X________
Secretary processing time (5 minutes per message) . X________
Labor rate for Secretaries: ($35 per hour / 60) . . X________
Total cost of sending messages per month per PC user ($73) . . . . =________

Additional Processing Costs:
Number of messages that need to be sent per month per person: (100) . . . . ________
Average number of people who need to be copied per message: (5) . . . . X________
% of messages that need to be sent overnight (5%) . X________
Federal Express Costs per message ($10) . . . . . . X________
Total cost of sending messages per month per PC user ($250 . . . . =________

E. COST SUMMARY:

1. Labor savings in phone tag - simple response desired ($66) . . . . ________
2. Labor savings in phone tag - verifying message received or read ($33) . . . . +________
3. Labor savings in sending messages ($875). . +________
4. Material costs in sending messages by mail ($500). . . . . +________
5. Additional savings in sending messages overnight ($250) . . . . +________
6. SUBTOTAL ($1,724) . . . . =________
7. Number of PCs connected to system (20). . . X________
8. TOTAL SAVINGS ($34,480) . . . . =________

GRAPHICS:

According to Computer Graphics World, presentations using visual aids were 43% more persuasive than unaided presentations. Persuasion has value and can have an impact on an organization's bottom-line. An increase of 43% in persuasion can result in a positive impact on an organizations bottom line. What if you could sell 43% more of your customers on buying your products resulting in a 43% increase in sales. A company with sales of $1 million and a ratio of net-earnings to sales of 8% could expect an increase of $430,000 in sales and an increase in net earnings of $34,400.

MANAGEMENT CONCERNS:

While office automation can be justified as seen above, there are some concerns that need to be addressed.

Three questions that a "management cost justification" should address are:

* How much is going to be saved?
* How accurate is the analysis?
* How soon will the results be achieved?

By increasing the efficiency of how people work in organizations, time can be saved. By investing the time-saved effectively, tasks can be added that contribute to the organization's bottom-line.

In situations where office people are working 60 hours per week, one can assume that this is approaching their limit. Any increase in work will probably necessitate an increase in staff. When situations such as this occur, the costs of hiring additional people needs to be addressed. These costs can become a real expense.

IMPLEMENTING OFFICE AUTOMATION PROJECTS:

While it can be seen that Office Automation can be cost-justified, in order to achieve the results it is important to have a successful implementation.

The following are some guidelines on how to successfully implement an Office Automation project:

1. **Plan ahead.**
Real office productivity comes form changing work processes and eliminating unnecessary steps, not just speeding up work.

2. **Involve top management.**
In order to maximum the implementation benefits, top management support should be obtained. This is important for office automation implementations that span organizational boundaries.

3. **Be selective.**
Identify one or two well defined tasks that are critical to your company's mission and make sure the objective is measurable. Don't give everyone computers at first. Target the 20% who can assure the success of your initial experiments. The other 80% will follow.

4. **Be patient.**
Learning curves need to be considered. Technology will change your organizations culture. Employees need time to adjust to different ways of doing work.

5. **Measure the benefits.**
Doing things faster is a change, but not necessarily a benefit. Monitor whether the technology alters behavior, assess whether that is good, and calculate the value of that change. Figure where the time went. Sometimes the greatest payoff comes from doing a better job, not from just reducing staff.

6. **Communicate.**
Tell your employees why you want to automate and get their help in doing the job. Changing behavior is a difficult task. However, when the advantages become understood and the value internalized, changes are easier to incorporate.

SUMMARY:

As a result of this paper and analysis, I hope that you find some techniques that will help you in your office automation justification efforts. I really believe that when properly addressed, Office Automation can be justified and make a positive contribution to an organization and the company's bottom-line. I hope that you would try the techniques. The HP field sales organization has access to additional Office Justification Worksheets that can also help you.

STRATEGIC IMPORTANCE OF RELATIONAL DATABASE

Orland Larson
Hewlett-Packard Company
Cupertino, California

ABSTRACT

It is generally understood that the enterprises most likely to excel in the future will be those that have recognized the importance of managing information as a major resource. These companies will be competing on the basis of the accessibility, accuracy, and timeliness of their information. One of the most important technologies used to improve the productivity, speed and flexibility of information management systems is RELATIONAL DATABASE.

This paper begins by addressing the issues and concerns associated with current non-relational application development environments. This will be followed by a review of the motivation for relational and an overview of the relational approach including Structured Query Language (SQL). The things to consider when choosing relational will be addressed including a review of the characteristics of an application that would be best suited for relational. This will be followed by a summary of the relational approach and a discussion of the things to consider when preparing to "go relational". The proprietary and third-party relational products available on HP computer systems will then be discussed followed by a preview of the future trends in relational technology. Finally, this paper will summarize the strategic importance of relational database technology for your organization.

INTRODUCTION

Database management systems, which first became widely available in the early 1970's, are, for many users, the single most important piece of software they will ever own. The implementations of the network and hierarchical models have provided developers with the tools that are generally providing users with solutions to complex information management problems. However, there is a price to pay for using either of these nonrelational approaches. Because of the navigational nature if these two models, database designers spend a great deal of time predefining data relationships only to find that users' data requirements are changing dynamically. These changes in user requirements cause modifications to the database structure and, in most cases, the associated application programs. This results in an excessive amount of time maintaining applications that are inflexible and fail to meet user needs. In addition, the complexity of accessing data limits the productivity of the programmer who is forced to think and code at a low level of structural detail.

Business professionals are frustrated by the limited access to information that they know exists somewhere in the database. Their business environment is changing dynamically, and they feel MIS should keep up with these changes. They also lack powerful inquiry facilities to aid in the decision-making process, which would allow them to ask anything about any data residing in that database.

THE MOTIVATION FOR RELATIONAL

Dr. Edgar F. Codd, considered to be the originator of the relational model for databases, noted when presented the 1981 ACM Turing Award that the most important motivation for the research work resulting in the relational model was the objective of providing a sharp and clear boundary between the logical and physical aspects of data base management (including data base design, data retrieval, and data manipulation). This is called the data independence objective.

A second objective was to make the model structurally simple, so that all kinds of users and programmers could have a common understanding of the data, and could therefore communicate with one another about the database. This is called the communicability objective.

A third objective was to introduce high-level language concepts to enable users to express operations on large chunks of information at a time. This entailed providing a foundation for set-oriented processing (i.e., the ability to express in a single statement the processing of multiple sets of records at a time). This is called the set-processing objective.

Another primary motivation for development of the relational model has been to make data access more flexible. Because there are no pointers embedded with the data, relational programmers do not have to be concerned about following pre-defined access paths or navigating the database, which force them to think and code at a needlessly low level of structural detail.

RELATIONAL DATABASE DEFINED

The relational database model is the easiest to understand - at least at the most basic level. In this model, data are represented as a table, with each horizontal row representing a record and each vertical column representing one of the attributes, or fields, of the record. Users find it natural to organize and manipulate data stored in tables, having extensive familiarity with tables dating from elementary school.

The Table, or two dimensional array, in a "true" relational database is subject to some special constraints. First, no row can exactly duplicate any other row. (If it did, one of the rows would be unnecessary). Second, there must be an entry in at least one column or combination of columns that is unique for each row; the column heading for this column, or group of columns, is the "key" that identifies the table and serves as a marker for search operations. Third, there must be one and only one entry in each rowcolumn cell.

A fourth requirement, that the rows be in no particular order, is both a strength and a weakness of the relational model. Adding a new record can be thought of as adding a row at the bottom of the table; hence, there is no need to squeeze a new row in between preexisting rows as in other database structures. However, to find a particular row, the entire table may have to be searched.

There are three kinds of tables in the relational model: base tables, views, and result tables. A base table is named, defined in detail, filled with data, and is more or less a permanent structure in the database.

A view can be seen as a "window" into one or more tables. It consists of a row and/or column subset of one or more base tables. Data is not stored in a view, so a view is often referred to as a logical or virtual table. Only the definition of a view is stored in the database, and that view definition is then invoked whenever the view is referenced in a command. Views are convenient for limiting the picture a user or program has of the data, thereby simplifying both data security and data access.

A result table contains the data that results from a retrieval request. It has no name and generally has a brief existence. This kind of table is not stored in the database, but can be directed to an output device.

THE RELATIONAL LANGUAGE

The defacto industry standard language for relational databases is SQL. SQL is pronounced "SEQUEL" and stands for Structured Query Language. This name is deceiving in that it only describes one facet of SQL's capabilities. In addition to the inquiry or data retrieval operations, SQL also includes all the commands needed for data manipulation. The user only needs to learn four commands to handle all data retrieval and manipulation of a relational database. These four commands are: SELECT, UPDATE, DELETE and INSERT.

The relational model uses three primary operations to retrieve records from one or more tables: select, project and join. These operations are based on the mathematical theories that underlie relational technology, and they all use the same command, SELECT. The select operation retrieves a subset of rows, that meet certain criteria, from a table. The project operation retrieves specific columns from a table. The join operation combines data from two or more tables by matching values in one table against values in the other tables. For all rows that contain matching values, a result row is created by combining the columns from the tables, eliminating redundant columns.

The basic form of the SELECT command is:

```
SELECT     some data (column names)
FROM       some place (table names)
WHERE      search conditions (if any) are to be met
```

In some instances WHERE, may not be neccessary. Around this SELECT..FROM..WHERE structure, the user can place other SQL clauses in order to express the many powerful operations of the language like the GROUP BY clause or the HAVING clause.

In all uses of SQL, the user does not have to be concerned with <u>how</u> the system should get the data. Rather, the user tells the system <u>what</u> data is needed. This means that the user only needs to know the meaning of the data, not its physical representation, and this feature can relieve the user from many of the complexities of data access.

The data manipulation operations include UPDATE, DELETE and INSERT. The UPDATE command changes data values in all rows that meet the WHERE qualification; the DELETE command deletes all rows that meet the WHERE qualification; the INSERT command adds new rows to a table.

When retrieving data in application programs, it is important to remember that SQL retrieves sets of data rather than individual records and consequently requires different programming techniques. There are two options for presenting selected data to programs. If an array is established in the program, a BULK SELECT can retrieve the entire set of qualifying rows and store them in the array for programmatic processing. Alternatively, it is possible to activate a cursor that will present rows to programs one at a time.

SQL has a set of built-in, aggregate functions. Some of the functions available are COUNT, SUM, AVERAGE, MINIMUM, and MAXIMUM. They operate on a collection of values and produce a single value.

In addition to commands for data retrieval and modification, SQL also includes commands for defining all database objects. The data definition commands are CREATE, ALTER and DROP. The CREATE command is used to create base tables, views, indexes and authorization groups; the ALTER command provides for the expansion of existing tables; the DROP command deletes a table, view, index and group. One of the most powerful features of SQL is its dynamic data definition capability. This function allows the user to add tables, columns, views, indexes and groups to the database without unloading and reloading existing data or changing any current programs. More importantly, these changes can be made while the databases are in use.

WHEN TO CHOOSE RELATIONAL

The choice of the "correct" database management system must be based on the environment in which the database will be used and on the needs of the particular application. The key feature of relational technology is that it allows for maximum flexibility, and will probably be the choice for many new applications.

The relational approach should be selected when the application has a large number of data relationships or when the data relationships are unknown or changing dynamically. The relational approach provides the needed flexibility to establish relationships at the time of inquiry, not when the database is designed. If the application has unknown or incomplete data specifications, which is usually the case in a prototyping environment, then a relational database may be preferable. If the application requires a quick turnaround, the quick design and implementation capabilities of a relational database can be important. The ability to handle ad hoc requests is a definite strength of the relational model as is the ability to extract data for use in a modeling, forecasting, or analytical framework.

Some questions that should be asked when deciding whether or not to "go relational" are listed below.

* Does your company have an excessive backlog of applications to be developed, including an invisible backlog?

* Are your programmers spending too much time maintaining applications caused by changing data requirements or relationships?

* Are your programmers spending an excessive amount of time writing code to navigate through a nonrelational database?

* Do your users' requirements for information change dynamically?

* Do your users feel restricted by a nonrelational database?

* Would your users find it natural to organize and manipulate data in tables? For example, are they currently using spreadsheets?

* Is your company moving towards a distributed database environment?

If you answered yes to more than one of the above questions, you should seriously consider taking advantage of relational database technology.

RELATIONAL HIGHLIGHTS

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 to the level of problem solving.
* Data independence is ensured and minimizes maintenance of programs.
* Data access is automatically optimized as the DB structure changes.
* The DBA has unprecedented power and control over the database.
* New systems are implemented much faster.
* Relational databases provide a cost effective, powerful solution.

It is to the advantage of most dataprocessing management to learn to use this technology creatively and to manage it effectively.

PREPARING FOR RELATIONAL

There are several things to consider when going to a relational database environment. The first thing to be aware of are the additional computing resources required to effectively support this technology. For example, the dynamic capability of the relational approach and the intelligence built into the relational software usually requires additional CPU cycles and memory.

Training of the database administrators in database design and data modeling is something that should not be overlooked when developing the relational application environment. The programmer should also be trained in the use of SQL and associated application development tools.

Performance is usually brought up as an issue when discussing relational and often depends on the maturity of the optimizer software which is built into the relational software. The optimizer is used to determine the most efficient way of accessing the database. The Database Administrator (DBA) plays a major role in monitoring and improving performance by creating and dropping indexes when appropriate. The DBA can also elect to use "clustering" or "keeping like data together" which affects performance by reducing the number of times a disc is accessed. Fortunately, Hewlett-Packard's Precision Architecture Computers are providing a high performance platform for relational database applications.

The command-driven nature of SQL may be difficult for some users to understand because they usually have to know the names of the tables and columns in order to properly construct a SQL command. The user may prefer a much more "friendly" interactive menu-driven interface such as Hewlett-Packard's ALLBASE/QUERY on MPE/XL or HP VISOR on MPE/V systems.

Security of the data resources is usually very important. The DBA has the capability to implement some very comprehensive security schemes. In addition, to ensure data integrity, logging of transactions against the database is mandatory and provides for the automatic recovery of the database.

If compatibility with SQL in an IBM environment is important, HP's SQL is very similar. The user and programmer interface is essentially the same; however, there are some DBA functions which are system dependent. Both of these products, as well as others on the market, are converging on industry standard interfaces.

RELATIONAL PRODUCTS IN THE "HP WORLD"

Hewlett-Packard is strategically committed to relational database technology. HP is deeply involved in the various standards committees such as the American National Standards Institute (ANSI) and the international standards oganization called X/OPEN. HP's development team is also aware of the importance of SQL and DB2 from IBM and Codd's rules for a fully relational database.

Significant investment is being made to broaden HP's relational offering and to integrate additional products with this key technology The first of these integrated products, ALLBASE/4GL and ALLBASE/QUERY, were introduced at the recent INTEREX meeting in Sweden.

ALLBASE/4GL (previously HP TODAY) is a highly integrated fourth generation environment for developing and maintaining transaction processing applications. Its outstanding qualities include its excellent prototyping facilities and complete integration of its component tools (screen painter, report writer and high level logic commands) resulting in improved programmer productivity.

ALLBASE/QUERY (previously HP VISOR) is an easy to use, terminal-based query and reporting tool for HP SQL end users. It combines a simple forms-based interface with function key operation to allow end users to perform queries and generate their own reports.

The following table shows the relational database products that are currently supported by Hewlett-Packard. HP VISOR will be changed to ALLBASE/QUERY and HPTODAY will be changed to ALLBASE/4GL on HP-UX systems in November 1988.

HEWLETT-PACKARD'S PROPRIETARY RELATIONAL PRODUCTS

MPE V	MPE XL	HP-UX
HP SQL	HP SQL	HP SQL
HP VISOR	ALLBASE/QUERY	HP VISOR
	ALLBASE/4GL	HP TODAY (4GL)

The following table shows the relational database products that are supplied by HP's third party independent software vendors.

HEWLETT-PACKARD THIRD PARTY RELATIONAL PRODUCTS

MPE V	MPE XL	HP-UX	MSDOS
RELATE/3000	ORACLE (2H88)	ORACLE	ORACLE
	INGRES (1H89)	INGRES	RBASE
	RELATE/3000	INFORMIX	INFORMIX
		UNIFY	

Hewlett-Packard is currently evaluating other relational database and 4GL vendors with an interest in porting their products to HP computer systems.

FUTURE TRENDS IN RELATIONAL TECHNOLOGY

Relational database is becoming a 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.

Within the next few years, distributed database will provide users with the ability to access data transparently and simultaneously across multiple relational databases on computers in a network. Information will be viewed as one logical database even though it may be fragmented across tables, databases and different vendors' computers. Efficient distributed database will only be possible with relational database. The continued development, growth and maturity of relational databases has made new technologies such as distributed database an extremely important direction for Hewlett-Packard in the future.

STRATEGIC IMPORTANCE

Relational databases can significantly 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 very 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 also results in a significant reduction in the maintenance of applications.

Overall, the use of relational technology can increase the MIS professionals' 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 future.

SUMMARY

Relational technology can have a profound effect on the way organizations operate. In short, the use of relational databases, within the correct environment, can help turn the computer into the effective tool most managers need to run their organizations effectively. It would be advantageous for your company to become very familiar with relational technology and to begin using this powerful tool. Paul Hessinger, Vice President of Research and Technology at Computer Task Group, said in a recent Computerworld article, "Successful, insightful users of relational DBMS have used 'going relational' as a catalyst for refining and in some cases redefining their entire approach for building and delivering information systems."

REFERENCES

Codd, E.F., "A Relational Model of Data for Large Shared Data Banks," CACM, 13 6,(June 1970) pp. 377-387.

Codd, E.F.,"Relational Database: A Practical Foundation for Productivity," CACM, 25 2,(February 1982) pp. 109-117.

Date, C.J., An Introduction to Database Systems, Addison-Wesley, 1977.

Date, C.J., An Introduction to Database Systems Vol II, Addison-Wesley, 1983.

Larson, O.J.,"Relational Database: How Do You Know You Need One?", Proceedings 1987 North American Conference of Hewlett-Packard Business Computer Users,Las Vegas, Nevada, September 20-25, 1987.

Schussel, George,"Relational Database-Management Concepts, "Proceedings 1988 Database and Fourth/Fifth Generation Language Symposium, Washington, DC, March 21-23, 1988.

________,Relational Technology: A Productivity Solution, Hewlett-Packard Co., Computer Systems Division, Cupertino, Ca., 5954-6676, January 1986.

________,SQL/Data System for VSE: A Relational Data System for Application Development, IBM Corp. Data Processing Division, White Plains, N.Y.,G320-6590,Feb 1981.

TITLE: Performance Tools Selection and Design

AUTHOR: Paul Primer

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 2080

PC Integration with HP AdvanceNet

David R. Schwaab

Hewlett-Packard Company

3404 East Harmony Road

Fort Collins, Colorado 80525

Personal computers have become common in the workplace because they offer increased productivity with specialized applications, user independence, and desktop power. They are found on the desks of executives, programmers, and clerical personnel alike. The use of PCs to improve productivity has resulted from technological progress and the need for more computing power in the hands of users. Many companies are tolerating segregation between their PCs, and between their PCs and their departmental computers. Other companies have provided terminal connections to departmental computers with terminal emulation and file transfer for their PCs via point-to-point RS-232 links. These companies now see that further growth in productivity will come from increased communication between their desktop PCs and departmental computers.

These companies now face the challenge of integrating these PCs into the larger information systems of their organizations. This effort is both a challenge and a tremendous opportunity to gain the benefits that come from increased communication, network services, and the central administration of assets - computing hardware and peripherals, applications and data, and technical expertise. These benefits are increased productivity, improved quality, and lower overall costs.

This paper focuses on how PC integration is achieved with HP AdvanceNet to increase communication and productivity. The topics covered are:

- o The advantages and disadvantages of PC integration using a LAN
- o Network cabling
- o Network links
- o Network services software

The Advantages and Disadvantages of PC integration Using a LAN

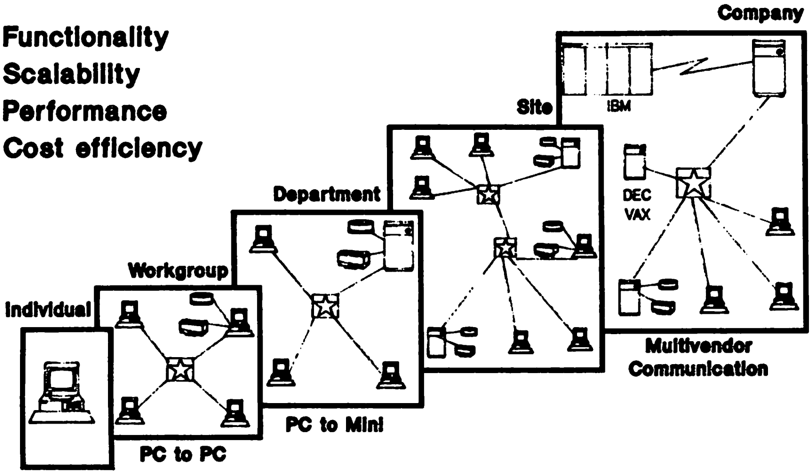
An HP AdvanceNet LAN is the most effective method to achieve PC integration because it enables the PC to become the entry point into an entire corporate computing environment at whatever level the user requires, either the local workgroup, department, site, or corporate processing level. Rather than connecting PCs to departmental computers as terminals, or to other PCs using a PC-only LAN with point-to-point connections to a departmental computer, an HP AdvanceNet LAN integrates every PC with all other computers on the network, from PCs to corporate mainframes.

The LAN provides this interconnectivity, with shared data and resources residing on multiple CPUs, and allows multivendor connectivity via terminal access and file transfer. The LAN also enables you to use distributed applications across multiple CPUs. With HP AdvanceNet, the PC user has access to all of these capabilities with only one, consistent network user interface.

Using an HP AdvanceNet LAN for PC integration allows for completely scalable growth from an individual PC all the way up to a corporate mainframe. It enables you to add users and servers when and where they are needed. People may work individually, as a local workgroup with a PC and/or HP 3000 server, with multiple workgroups integrated with a departmental computer, as an integrated site with many interconnected departments, and finally, with many integrated sites as a company.

PC Integration with HP AdvanceNet

- Functionality
- Scalability
- Performance
- Cost efficiency



The price/performance ratio is generally better with a LAN than with point-to-point connections. A LAN provides better file transfer speed with only a little more HP 3000 processing overhead. Also, more users don't degrade each others' individual performance because they each have their own processing power. An industry move is on to take advantage of this distributed computing environment by developing distributed applications that minimize data transfer between processes, e.g., concentrated database access. Because the price per PC MIP is coming down faster than that of minicomputers, it is easy and cost effective for users to add more PCs to a network to get more processing power.

A LAN is also the most cost effective method for integrating PCs with HP 3000s, other PCs, and minicomputers. The initial incremental cost of LAN connections compared to point-to-point connections is attractive when considering the additional functionality, connectivity, and scalability a LAN provides. A LAN protects users' investments in new and existing PCs, peripherals, software and data, and paves the way for their use of future LAN functionality including new distributed applications.

The disadvantage of using a LAN to integrate PCs is the initial incremental cost of the network components and time required to install and manage the network above point-to-point connections. This initial incremental cost varies by HP 3000 processor size and the number of PCs to be connected, but generally averages about $500 per PC. When considering the advantages of a LAN and the value of users' time acquiring, creating, and communicating information, this initial incremental cost is marginable. The functionality provided over an HP AdvanceNet LAN is superior to point-to-point connections because RS-232 links cannot provide the interconnectivity, services, and scalability of a LAN. The advantages of using a LAN to integrate PCs outweigh the disadvantages.

The Network Components

Three basic network components are used to integrate PCs into an HP AdvanceNet LAN. They are:

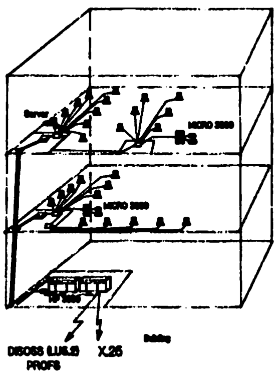
- o the network cabling
- o the network links (the PC interfaces on that cabling)
- o the network services software.

Network Cabling

The wiring infrastructure of a building or campus is the foundation upon which every well-conceived network design must rest. If this foundation is inadequate or missing, it will be impossible to build an effective, flexible and manageable network solution.

Wiring is a Business Asset

- Wiring is the network foundation
- A strategic business decision
- HP provides all necessary guidelines, products and services



Every business environment has its own unique needs, characteristics and computing automation programs. The network must be versatile to meet the wide range of information needs in your organization, and be flexible so it can grow as those needs change. The network must provide connections to many vendors' systems to protect user investment, and solve complex communication problems.

To achieve all this connectivity, the network must be based on a comprehensive wiring foundation. Users need to be connected together in logical workgroups to share data and resources. Workgroups then must be connected to a site backbone to provide facility-wide communication. The office wiring system must be well defined and limited to a uniform medium to eliminate costly rewiring for computer moves, adds or changes.

Although the cost of network user interface hardware and software has been decreasing significantly, the cost of network cabling has remained relatively constant. When we also consider that building wiring has a life span two to six times greater than the equipment it connects, we realize that wiring, which was once a secondary network consideration, is of strategic importance. The choice of wiring media is one of the most important long-term decisions an MIS manager can make.

Recognizing the importance of cabling, HP has developed a complete set of communications wiring guidelines, products, and services, called *HP SiteWire*, to help customers with their wiring decisions. HP SiteWire adheres to an open, proven multi-vendor wiring foundation that follows the guidelines of an emerging industry standard, EIA (Electronic Industries Association) TR-41.8. Adherence to standards ensures that the wiring system will provide multi-vendor compatibility and lasting value.

The framework for HP SiteWire is a wiring architecture based on this standard. HP's wiring architecture addresses the needs of any physical environment and also provides a way to implement networks in a controlled, step-by-step manner. The wiring architecture is based upon a distributed star topology compatible with existing telecommunication systems. The distributed star topology provides easy network cable administration and flexible growth. Such a structured wiring system simplifies network management and reduces the cost of adding and moving network users from an average of $1,000-$1,500 to $200-$300 each.[1]

The architecture includes the use of a thin or thick coaxial backbone cable running horizontally and/or vertically through a building with horizontal subnets of unshielded twisted pair or coaxial cabling, depending on the environment. With unshielded twisted pair wiring, the wiring system that once supported only the phone system can now be used to also support data applications.

There are many benefits to the use of unshielded twisted pair wiring over shielded twisted pair and coaxial cable. Unshielded twisted pair cable costs less and is easier to install. "Unshielded cables offer much more flexibility. They are of small cross-section, making installation easier and requiring less space in ducts and satellite closets. They also currently support high-speed data transfer..."[2] In addition, the existing unshielded twisted pair voice system in a building may also be able to support data as well, virtually eliminating the large cabling cost component of a LAN. As network hardware and software costs continue to decrease compared to the relatively constant cost of wiring, the cost of adding a LAN to an appropriately cabled building will continue to drop substantially from that of a building requiring cabling installation.[3]

---

1 "Structured Distribution Systems", Local Area Communications (Gartner Group, Inc., January 30, 1987), Key Issues K-CBL-319.1

2 "Shielded vs. Unshielded Cable", Local Area Communications (Gartner Group, Inc., August 21, 1987), Technology T-WIR-377.1

3 "Data Switching: Price Trends", Local Area Communications (Gartner Group, Inc., September 5, 1986), Scenarios S-LAN-274.1

An unshielded twisted pair cabling system can support voice as well as data transmission, eliminating the need for two separate voice and data cabling systems. A coaxial cabling system for data requires a separate cabling system for voice. Using unshielded twisted pair wiring can save the cost of the additional system. It is less expensive to pull two parallel, voice and data systems at the same time than two separate systems. Whenever cabling must be installed, the cost of labor is the major component. Approximately 60% [or more] of the cost of wiring a new building is attributable to labor for installation.[4] In many wiring quotes we have seen, the cost of labor has been as high as 80%-90% of the total quote. If walls must be opened or if new wiring troughs or conduits must be installed because the existing ones are full, the cost of installation increases dramatically. Because unshielded twisted pair is smaller, additional costs such as these may be avoided. An additional major benefit to using unshielded twisted pair wire over coaxial or shielded cable is that it often already exists, requiring *no* cable installation.

Testing Existing Wiring

When a customer wants to utilize an existing unshielded twisted pair wiring system whose condition is unknown, a wire test may be necessary. The wire specifications to support higher speed (1Mbps and 10Mbps) data communications is much stricter than that required to support voice communications.

HP offers a unique service, called *HP WireTest*, which strengthens HP's StarLAN offering by providing customers with an evaluation of the suitability of their existing unshielded twisted pair wiring for a StarLAN network. The availability of HP WireTest

---

4 "The Cost of Network Ownership" PC Netline (Hyatt Research Corp., July/Aug. 1987), p.1.

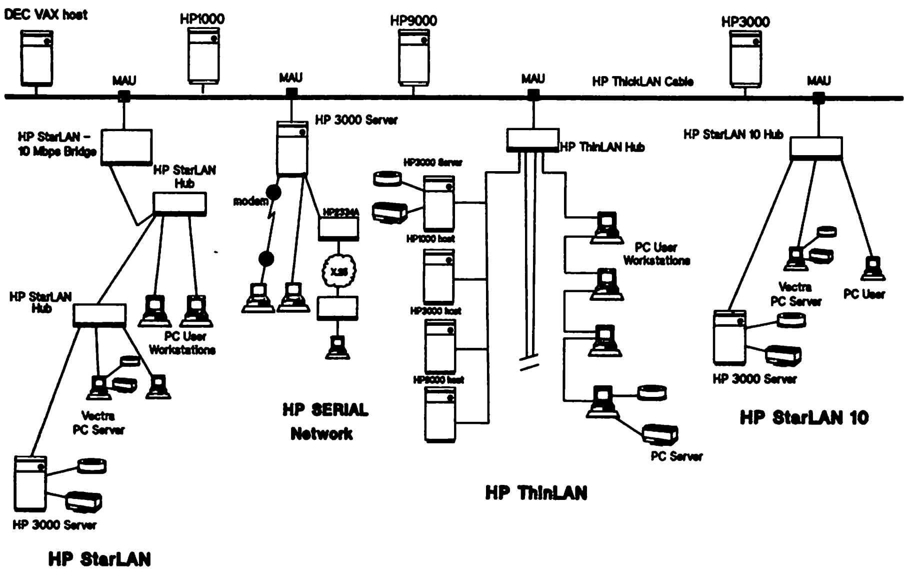
demonstrates HP's commitment to the development of complete customer networking solutions, and the cost is very low when compared to the cost of installing new wire.

Network Links

A network link is the interface by which a PC is connected to the network cabling. The network links can be used with the cabling for some networking applications independent of the network services software, and are therefore sold separately from the services. The network services software can also be used over any of the network links, offering an extremely flexible and scalable architecture. Every link can be integrated into a single, enterprise-wide HP AdvanceNet LAN to provide all users with the access and services they need.

HP AdvanceNet offers four network links. *HP StarLAN* and *StarLAN 10* allow customers to use existing telephone wiring to support their office data communication needs HP StarLAN and StarLAN 10 are compatible and complementary, offering flexibility in meeting users' data communication needs, while preserving the investment in their wiring asset. *HP ThinLAN* is available for those customers who have coaxial cable already installed for PC connectivity, and the *HP SERIAL Network* link provides a remote, asynchronous (point-to-point) link to network services on an HP 3000.

Network Links



o HP StarLAN

HP StarLAN is a 1 Mbps LAN link using unshielded twisted pair wiring. This wiring often already exists, running parallel with the customers telephone wiring. The link supports HP Vectra, IBM™ PC/XT™/AT™, IBM PS/2™ Models 25 and 30 personal computers, and the Micro 3000 and HP 3000 Series 37. Personal computers on a StarLAN

IBM, AT are registered trademarks of International Business Machines Corporation.
PS/2, XT are trademarks of International Business Machines Corporation.

network can communicate with other PCs and minicomputers that are on StarLAN 10, ThinLAN or ThickLAN networks via a StarLAN bridge.

o HP StarLAN 10

StarLAN 10 is a 10 Mbps LAN link, using unshielded twisted pair wiring, for HP Vectra, IBM PC/XT/AT, IBM PS/2 Models 25 and 30, HP Touchscreen (North America only) personal computers, and HP minicomputers. HP StarLAN 10 will allow customers to use high-speed applications and workstations, such as 80386 processor-based personal computers, without the need to install new building wiring in many cases. It is a flexible network solution for integrating PCs and minicomputers in complete office automation solutions. It is especially useful in business office environments that require a large number of nodes and experience heavy network traffic.

o HP ThinLAN

HP ThinLAN is a 10 Mbps thin coaxial cable LAN link, supporting HP Vectra, IBM PC/XT/AT, IBM PS/2 Models 25 and 30, and HP Touchscreen PCs, as well as HP minicomputers. It is especially useful where coaxial cable is already installed, or in engineering and manufacturing environments.

o HP SERIAL Network

The HP SERIAL Network link provides an asynchronous connection to HP 3000 computers for HP Vectra PCs, HP Touchscreen PCs, IBM PC/XT/AT, and IBM PS/2 Models 25, 30, 50, 60 and 80. This connection allows remote PC access to shared peripherals and PC files residing on HP 3000 servers, distributed applications, terminal emulation, and network file transfer (NFT). HP Vectra and IBM PCs can also make this connection via back-to-back HP 2334A multiplexers over an X.25 network.

Mixed Link Networks

As specified by HP SiteWire guidelines for the greatest growth potential with minimal geographic restraints, ThinLAN or ThickLAN coaxial cable can be used as a backbone for a site-wide LAN. HP StarLAN, StarLAN 10, and ThinLAN sub-networks can be attached to a backbone cable to increase distances, the number of network nodes, and to provide intercommunication for all of the computers on these LANs.

HP's support of all these links provides customers with complete flexibility to meet specific and multiple connectivity needs for the least cost.

Network Services Software

The network services software runs on the PCs that are connected via the hardware links. This software provides the network services which allow the PCs to communicate with each other and with HP 3000, HP 1000, HP 9000, and DEC™ VAX™ computers, and enable communication with corporate mainframes. The *HP OfficeShare Family of Networking Software for PCs* is the group of software which provides these network services to PCs. The HP OfficeShare software supports the HP Vectra Family of PCs, the HP Touchscreen, IBM PC/XT/AT and PS/2 Model 25, 30, 50, 60, and 80 personal computers, preserving users' investments in PCs.

Like all HP AdvanceNet products, OfficeShare is based on the International Standards Organization OSI networking model. OfficeShare provides compatibility with IEEE 802.3 industry standards, and employs de facto standards such as Microsoft™ Networks,

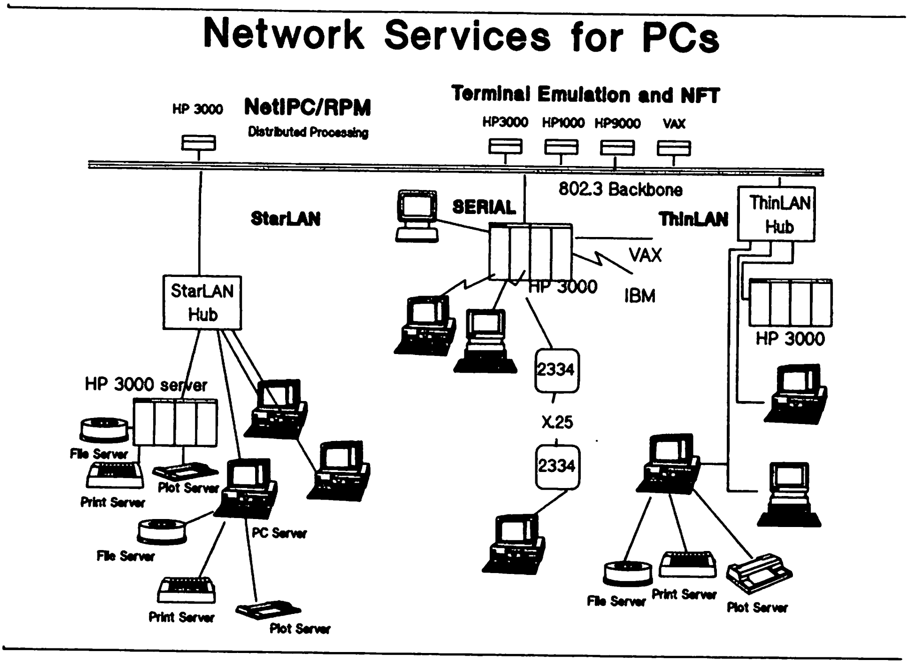
---

DEC, VAX are trademarks of Digital Equipment Corporation.

Microsoft is a registered trademark of Microsoft, Inc.

allowing applications written to the MS-Net™ interface to function on the network. This support of industry standards protects your investment and ensures the widest selection of growth options available. The OfficeShare products also provide a single, consistent PC user interface to all network services, providing easy user integration into an enterprise-wide LAN.

Resource Sharing is the network software which enables the HP 3000 to function as a server on a network. Resource Sharing is a component of *HP Business System Plus*, an integrated system of business applications for networked PCs and HP 3000s.



MS-Net, MS-DOS are registered trademarks of Microsoft, Inc.

The network services software provides:

o Peripheral Sharing

The PC server software and the HP 3000 server software enable users to enhance communication, increase productivity, and reduce costs by sharing files and peripheral devices such as discs, printers, and plotters. Users can store and retrieve files on discs connected to the servers as if they were local. There are no new commands to learn; users only need to know the MS™-DOS device letter (C:, D:, etc.) of the files they want to access. Network discs are structured using the MS-DOS hierarchical directory structure and can be tailored to suit users' needs. Normally, the network manager creates a directory for each user; users can easily create their own subdirectories. Users can share files among all types of HP and IBM PCs, using passwords and three access levels (read, write, and create) to provide the proper security for their files. PCs, HP 3000s, and combinations of both may function as servers to network users. Printers and plotters connected to a network server may also be shared as if they were locally connected devices. Multiuser access to printers and plotters is transparent through the use of a spooler.

o Terminal Access to HP 1000, HP 9000, and DEC VAX Hosts

Additional connectivity is offered to PC users through terminal access to HP3000, HP 1000, HP 9000, and DEC VAX computers. PCs on a LAN can serve as terminals to many different computers, thus saving money by reducing equipment investment and saving time by providing multihost connectivity from a single workstation. This virtual terminal capability provides block mode terminal emulation to the HP 3000, and telnet service to

---

MS is a registered trademark of Microsoft, Inc.

the HP 1000, HP 9000 (all models except the 500) and DEC VAX computers. The terminal emulation capability is accessed by the user via the HP Terminal Program.

o Network File Transfer

Network File Transfer (NFT) allows better communications and data exchange between PC users and their network computing environment. NFT provides network file transfer between PCs and HP 1000, HP 3000, HP 9000 and DEC VAX hosts. The NFT service is accessed by the PC user with the DSCOPY command, and may also be accessed programmatically.

o PC-to-PC Communications

OfficeShare implements a NetBIOS application interface supporting specific NetBIOS applications for PC-to-PC communication. NetBIOS support for tested third party applications allows users access to key multi-user applications including modem-sharing and electronic mail. Currently, these listed communications applications are *Hayes Smartcom II*, and *Network Courier*.

o MS-Networks and MS-DOS 3.X Windows Support

By employing MS-Networks, popular MS-DOS application programs can be used on the network, protecting users' investments in application software. Support of MS-DOS 3.X Windows allows for increased productivity by permitting users to run applications in windows for easier movement and interaction between programs.

o Distributed Application Development

Distributed application development software provides programmatic tools which enable application developers to write integrated, distributed applications utilizing the processing power of PCs and HP 3000s.

Distributed applications are extremely important today. Applications which transparently run on two or more computers and communicate over the network can make the most efficient use of distributed systems and provide timely information when and where it's needed. Using distributed application development tools, application developers may now easily integrate PCs - the most widely distributed, and often untapped, level of computing power across an enterprise-wide network.

HP offers two distributed applications development software tools, *Cooperative Services* for applications requiring HP 3000 TurboIMAGE database access from a PC on the network, and *NetIPC/RPM* for PC and HP 3000 task-to-task communication over the network.

Tailored for the development of commercial applications, Cooperative Services allows PC applications to programmatically access and update HP 3000 IMAGE and MPE file data, or call HP 3000 procedures. Cooperative Services simplifies development by eliminating the need for software designers to code in lower-level network intrinsics or provide parallel coding on the HP 3000.

The NetIPC/RPM Development Package for PCs allows developers to write distributed applications using PCs on a network. The software package is a set of Network InterProcess Communication (NetIPC) and Remote Process Management (RPM) program intrinsic libraries that provide standard communication between concurrent peer processes on PCs and HP 3000s, and the ability for PCs to start and manage processes on HP 3000s. HP's distributed applications such as AdvanceMail, Information Access, and Cooperative Services use NetIPC for LAN communications.

The NetIPC/RPM Development Package for PCs contains program libraries for use in Microsoft C and Lattice C programs. Other languages, such as Pascal, can call C routines, providing flexibility in the programming languages developers may use. The libraries are linked by the programmer to the PC application being developed, and called by the application to provide the network communications required. On the HP 3000, concurrent processes are developed using NetIPC/3000 and RPM/3000. NetIPC/3000 is provided with the HP 3000 network link product, such as the ThinLAN/3000 LAN Link, and RPM/3000 is part of NS/3000.

To use the distributed application, users only need the appropriate HP OfficeShare PC link (StarLAN 10, StarLAN, ThinLAN or SERIAL Network), and network configuration software. They do not need the network services software. Applications using the NetIPC/RPM libraries may, however, be run concurrently with network services software, allowing users to take advantage of both the application and services such as HP Vectra and HP 3000 servers.

Conclusion

PC integration goes beyond tracing a connective media from PCs to departmental computers. It is the seamless integration of the services and capacity, data, and applications residing on all computers interconnected throughout an entire company. This integration is the basis for further productivity gains at the desktop.

HP AdvanceNet integrates PC users into scalable, low-cost networks that provide enhanced PC-to-mini integration, multi-vendor communication, and PC-to-PC communications, for increased productivity, improved quality, and lower overall costs.

PC integration is a key part of HP AdvanceNet, Hewlett-Packard's long-term strategy for providing high-quality networking solutions for HP and non-HP computers. This strategy is based on the International Standards Organization OSI networking model and includes compatibility with industry standards, such as IEEE 802.3, and de facto standards such as Microsoft Networks, to ensure lasting value. HP AdvanceNet signifies a commitment by Hewlett-Packard to provide powerful yet easy to use networks with a well-supported growth path.

TITLE: Moving HP 3000 OA Users to PCs: What's Involved?

AUTHOR: Carol Agne

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 2082

Interex
North American Conference of Hewlett-Packard
Business and Technical Computer Users

Managing a small office local area network - HP Officeshare

Belinda Yung-Rubke
Product Manager
Colorado Telecommunications Division
5070 Centennial Boulevard
Colorado Springs, Colorado 80919

Introduction

In today's business world, efficiency in a communication network plays a key role in a company's success. In the office environment, the way information and resources are shared determines the productivity of a work group. Local area networks have played a large role in enhancing the flow of information in department work groups. While vendors of small office LANs have continually made installation and implementation easier, the basics for managing such networks have remained the same.

There are four phases to managing a small office network. They are planning, installation, user training and network maintenance. The examples I will be using came from my experiences with HP Officeshare.

Phase I - planning the network

Planning a network consists of three parts. They are network design, cable plant mapping, and naming and addressing selection for user nodes.

Network design includes selecting a network and peripherals and choosing applications from those supported on your particular LAN. The basis for selecting a network should be determined by the applications which are chosen or which need to supported. Applications include shared peripherals, application software and databases. At the same time that effort is being concentrated on designing your network around applications needs, it is necessary to identify who the network users will be. Logistics are a large part of designing your network, including not only who network users are and where they're located, but also selecting a centralized, easy-to-access location for the server and peripherals.

Cable plant mapping becomes the next important step in planning your network. Locations of users and servers play a large part in this process because most networks have distance restrictions. For example, the maximum length for each HP Thinlan segment is 185 meters. Because this paper is focusing on SMALL office networks, mapping out the cable plant should include visually inspecting the area where the cables need to run. For users in close proximity, the cable may not need to go into the ceiling. Instead, it can run along the floor as long as the exposed cable does not become a safety hazard.

The third step in planning your network is deciding on a naming and addressing convention for each node. Your facility systems administrator may be able to help you with a naming convention that is consistent with the rest of the site. Depending on the type of networking protocol used, you may need to select addresses for each node. Again, your system administrator will be able to provide the facility guidelines. If there are plans to connect the small office network to the site backbone in the future, then it is important that the addressing scheme is compatible with the site network. It is troublesome to go back to reset the addresses for the nodes. It is also a good idea to actually draw a detailed map of the network, including cable length, routing and node locations. This is a good way to communicate physical network layout to the installation person. You'll find it an even more useful tool if you're installing your network for yourself. At the same time you're mapping your network, start a list with user names, node addresses, and devices at the nodes. This network map and users list are basic documentation which will help you manage your LAN before, during and after implementation.

Planning doesn't stop after the network is installed and up and running. There is always room for expansion, adding nodes, shared peripherals and applications. The best way to manage changes to the network is to anticipate needs by staying familiar with your physical layout, your user needs and usage of the network.

Personal experience with HP Officeshare

In my case, the major reason the network was set up was so that I could gain experience, firsthand, in setting up and managing a LAN. As the product manger responsible for the LAN protocol analyzer business in marketing at my HP division, I had a vested interest in learning the LAN, from the ground up. Applications supported on the network include printer (HP LaserJet Plus) and plotter (HP 7550) sharing, file sharing (Memomaker and Lotus files) and application (Timeline) sharing. The network is HP Thinlan running Officeshare software. The network started out with no connection to the site backbone. The initial network had six user nodes and a single server node. All six users have access to both the plotter and printer. Three of the users have access to Timeline on the server. Since the LAN does not have any connection to the backbone network, all users connect to the mainframe computer environment (HP 3000s) through an asynchronous port. Initially, all nodes were Vectras.

My primary objective was to get the network up and running as soon as possible. I was anxious to try out the LAN protocol analyzer with real data to see how effective a tool it could be in helping manage the LAN (and to put it through its paces). As a result, I spent minimal time in network planning. However, I still used the simple planning guide in the HP LAN server installation guide and found it to be extremely easy to follow. Each node was assigned a name and an Internet Protocol (IP) address. Our facility system administrator for networks assigned our LAN a block of IP addresses and a department name. Our node names were xxxxxx.MARKETING.CTD where xxxxxx was the user's first name. Using first names turned out to be a little shortsighted. A better choice was using user's initials and internal phone extensions - BYR429.MARKETING.CTD. I eventually had to make server software configuration changes to rename all user nodes on a weekend. A little more planning would have alleviated this problem.

Phase 2 - installation

Installation consists of three steps: the cable network, the server and the nodes. Many small office networks are designed as user-installable. If, as part of network planning, it is apparent that part of the cabling for your network will reside in a ceiling or in floor trenches, it would be easier to have someone familiar with your site cabling implement your cable network for you. Again, before, during and after cabling installation, it is very important to physically map and update your original cable map. The map allows you to keep track of the size of the network, determine where best to add new user nodes and minimizes the time required to track down cable problems.

Server installation is the second step in the installation phase. Given a good manual, it should be relatively simple to install the hardware and to configure the server. After the server is configured, the next step is to bring up one user node. When in this phase of installation, pick a time when users being converted to network nodes are not using their PCs. This typically means after office hours or on weekends. Progress through user configurations one node at a time, testing all aspects of that user's configuration - access to shared peripherals and files, ability to load user software with no problem when booting. One distinct ease-of-use feature for users is to make network-user-software loading as transparent as possible. This can be done by adding the appropriate commands to an 'AUTOEXEC.BAT' file, so that it becomes an automatic part of the booting process.

Personal experience with HP Officeshare

The cable network for the small LAN which I managed was installed by our facilities maintenance personnel. Since the six users were not located in close proximity, the installer had to pay special attention to ensure that the entire LAN segment did not exceed 185 meters (the HP Thinlan IEEE 802.3 10base2 distance specification). In addition to wiring for the first group of users, he also put in drops for potential users. I kept a copy of the initial wiring map. The wiring map turned out to be difficult to keep updated because the layout of the area changed several times.

After the cable network was installed, the server and user nodes were set up. The server was booted first without any nodes attached to it. Then using the network-server software, the server was configured. In configuring the server, the first group of users as well as other potential users were defined so that I would not have to stop the server at some point in the future just to add a new name. Then, one workstation was configured to see if it worked. One by one, the rest of the users were configured and connected. The software and node hardware installation procedure was relatively simple as long as the manual was followed. As part of the user set up, commands were added to each user's 'AUTOEXEC.BAT' file to automatically load the LAN software when booting up.

Because the primary objective was to get a network up and running quickly (to test the capabilities of the LAN protocol analyzer), a detail was glossed over which blossomed into a bothersome task. The network-user software had been installed in the root directory. When, at a later point in time, the need to upgrade the network-user software arose, I ended up creating a batch file to delete the old LAN files and create a separate network-user software subdirectory for each node.

Total time spent setting up the server and the user nodes was about half a day. Early morning or late afternoon was the optimum time window to install the network-user software.

Phase 3 - user training

User training consists of three steps, preparatory (before the LAN is installed), novice (after initial LAN experience) and ongoing (when new features are added to the LAN. It is important to communicate to users the distinction between THEIR responsibilities and the LAN manager's responsibilities. Training users prior to using the network minimizes the number of start-up problems and errors. Novice training after users have used the network for a short period of time gives them the opportunity to ask informed and relevant questions. It is also a more appropriate time for the LAN manager to explain more in-depth about the LAN because users will relate and understand from their personal basis of experience. On-going training should be used to communicate new features on the LAN and to address any usage problems that may have surfaced. By staying in touch with your user base, you'll understand your LAN usage better and be a more effective manager.

Personal experience with HP Officeshare

Unfortunately, user training was not a priority when we installed and implemented our LAN. Coupled with the problem that most of the users did not have personal computer experience, the first two to three weeks were consumed with fire fighting. A lot of time was spent helping users set up their nodes so that they could access the shared peripherals on the network. There was more to making the LAN work than configuring user nodes. The number of repetitive questions from different users was amazing. It wasn't anticipated that I would be answering questions such as:

- I have just finished sending a file over to the server, why isn't it printing (or plotting)?
- Why is it taking so long ? (I had to explain about the single queue spooling of files to the server and had to educate users on how to check the output queue on the server to locate their position in this queue.)

Although training and educating users may sound like a lot of work and may take up much of your time, if the work is done up front, it saves time and trouble in the long run.

Phase 4 - using and maintaining the network

Personal experience with HP Officeshare

Using and maintaining the network involves many daily maintenance tasks. For example, in the start-up phase of the LAN which I managed, users were wondering why printouts were taking so long. It turned out the printer was out of paper. The users were not aware of the fact that checking the paper tray was a task which each and every user on the network owned. Each user should consider the shared peripherals as personal devices and treat the need for maintenance - supplying with paper, checking for errors, keeping a plotter stocked with fresh pens - as if the peripherals were connected solely to their PC. Another example was the shared plotter jamming when loading transparencies. Because of the single output queue, everything behind that one user's output was queued up, waiting until the problem with the plotter was noticed and resolved.

The network-server software does not support separate print and plot queues. Our user population included two graphics designers and two secretaries. Because of intensive output needs to both the plotter and the printer, it wasn't uncommon to find a needed PRN file queued up after 10 to 15 graphics files in the spooler. It typically meant a delay of an hour or more before the printout was in my hands. We have solved the single output queue problem by separating the printer and plotter to different servers on the same LAN.

When the network was first set up, we faced very tight budget constraints. As a result, the server only had a 20M byte hard disk. This is extremely marginal for six-user support of the LAN, particularly because four of our user population daily perform large file transfers through the server with a graphics application called Gallery. Because we limited our disk space to 20M bytes, the time it took to read a gallery file was extremely lengthy, tying up the server and the user's PC for long periods of time during the file transfer.

Massive cleanups of server disk space were necessary several times because users were not removing files from the public directory once they were done with them. Cleanups had to be done when the available disk space went down to 2M bytes. In one instance, a user working through the gallery output menu requested a continuous plot of 10 copies of the same graphic to the shared plotter. The plotting process was started when the user left work for the day, with the intention that it would run during low-usage network time at night. The next morning, the user found no output at the plotter. A check on the amount of disk space left on the server indicated that there were 0 bytes left. The server apparently was reading files faster than the plotter could process them. As a result, the 4M byte of space which had been free on the server filled up. Once filled, the server sat waiting for disk space to free up before it could proceed reading what the user-node PC was sending. The process became bottlenecked and quit.

There are several solutions to this problem. Optimally, load your server with as much disk space as possible. More simply, make use of the 'batch' file process for plotting a large number of graphics files continuously. A batch file allows a user to string together a series of drawings to be output continuously to the designated output device. This seems to work better with the HP Officeshare LAN. The server recognizes the end of each file (for each drawing) and begins processing the file as output, freeing up disc space for the next file which is being processed through the server.

We found that when we upgraded the network-server software, the problem had been addressed by allowing the server to continue to accept files from the user-node. However, once the disk space was used, the files were recognized as empty.

A hint is in order here. Regardless of the amount of disc space on your server, request that your users limit batch processing of graphics files to a maximum of 15 drawings. It wasn't unusual to find user-nodes on the LAN processing 40 to 50 drawings in a single batch file.

Problems that require troubleshooting

There were very few problems on the network that required serious troubleshooting. The most severe problem caused the server to lock up. It happened when one of the users tried to 'TYPE' a file located on the server. This is a simple, easy MS-DOS command to quickly look at a screen dump of a file. When the user tried the 'TYPE' command, nothing happened, with the end result being that no one else on the LAN could access the server. In the course of pinpointing the problem, a couple of potential problems were examined:

1. New nodes recently added to the network; and
2. A bad cable segment.

Even after the new nodes were taken off the LAN, the problem persisted. It seemed to be confined to a single node in the middle of the segment. Nodes further down (closer to the termination) were able to view the same file from the server. The network connection of the problem node was also examined very closely for physical cable problems. I also used the HP 4972A LAN protocol analyzer to watch the 'TYPE' transaction between the node and the server and observed that the server was sending out the file. In fact, the same frames were transmitted more than once, which is an indication that the server was not receiving an acknowledge from the node. At the same time, the node saw nothing of the file on the server.

The mystery was solved when someone mentioned that a new version of the transport software had been made available from our internal Information Systems department. It turned out the user had installed the new software without checking for backward compatibility with the version of software running on the server. A switch of the software solved the problem.

Tight control of software releases and coordinated software updates to both the server and user-nodes would have headed off the problem. This responsibility lies with the LAN manager.

Another problem requiring troubleshooting occurred after our department had been rearranged. A user tried to boot his node without success. It showed a strange boot-up error. Other user-nodes experienced the same problem. To make the problem even more obscure, each node had a different boot error. Since everything had worked the day before and the only element that had changed was the rearrangement of several offices (and the location of those user nodes), the next step was to isolate parts of the LAN with a terminator and reboot the isolated sections. That got to be very difficult after the first 2 nodes because there was not an updated version of the cable map available to assist in physically locating cable segments. The only recourse was to examine those cable segments that could be physically seen and traced by hand. An examination of cable segments located in the ceiling would have been the next logical step to pinpoint the problem. The culprit turned out to be an extra length of cable connecting a tee connector to a node card. The maintenance crew had discovered that the tee connector for one of the user-node PCs would not reach the back of the PC after relocating the user, so they added an extra cable segment from the tee to the card thus changing the impedance of the cable.

Another interesting fact about the network software was discovered by the HP 4972A LAN protocol analyzer. It was perceived as a 'problem' before close examination and investigation revealed that it was a factor of the boot process. The LAN protocol analyzer was continuously collecting network statistics even though utilization on this small LAN was low. During the initial stages of LAN implementation, the LAN protocol analyzer logged the occurrence of jabber frames on the network. These are frames longer than the allowable maximum (1514 bytes) on an 802.3 network. Since there were only six nodes on the network, the possible sources of the jabber frames were evident. After capturing the jabber frames with the HP 4972A LAN protocol analyzer, the content indicated that each node had sent out one jabber frame. We ran a few experiments and determined that it was all part of the LAN boot-up process.

Network expansion plans

Our LAN has expanded to ten users, with more potential users to be added yet. Usage has increased to the point that managing the LAN requires more time than I can put in. The realization that extensive preplanning will be needed to increase LAN productivity to its maximum possibility has had a sobering effect on everyone who uses the LAN as well as the LAN manager. Responsibilities for LAN management have been formally handed over to the 'marketing services' group within our marketing department. They have the charter for planning and implementing changes to the LAN which should keep us current with the latest which HP Officeshare has to offer. Together with the Information Systems department within our division, an expansion plan has been mapped out for the marketing network.

Our plan includes:

1. Two servers with one supporting the secretaries and graphics designers (5 people). The intent is to connect an HP eight-pen plotter (HP 7550) and a LaserJet plus to their server. This should help alleviate the single output queue problem for the rest of the user population who only require daily printouts on an infrequent basis. The remainder of the user population will be connected to a second server which will have access through its output queue to a LaserJet series II printer.

2. The number of LAN users has increased to 14. Since the current server software only supports 10 concurrent users, we will be updating to the latest revision of HP Officeshare software that supports 30 concurrent users.

3. Early this year, the department LAN was connected to the site-wide backbone through a Thinlan repeater. The marketing network is divided into two 185 meter segments with enough cable for future expansion beyond 14 users.

4. We would also like to eliminate asynchronous connections to the mainframe computer environment (HP 3000s), by using virtual terminal connections via the LAN.

5. The Information Systems department will also be working on a proposal to purchase a Mighty Mouse (HP 3000) for the next fiscal year for backing up the users on the marketing LAN. The computer will be located in the site-wide computer room and will be managed by the division Information Systems department. In the meantime, users can back up their disc via tape across the network.

Summary

In summary, I have discussed four phases in managing a small office network illustrated with my experiences as a LAN manager. To briefly review:

Phase 1. Planning the network includes selecting the services offered on the network (peripherals and applications), determining the number of users and the node devices to be connected, mapping out the cable network and network backup process. It also includes selecting addresses and a naming convention with assistance from your network management group. You also need to plan training sessions for users, allowing them the opportunity to use the LAN fully. This will also alleviate many start-up problems and help answer many questions. It is also useful to make up a schedule for the entire process from planning to installation to implementation.

Phase 2. The installation process includes the cable network, the server and peripherals, and the node software and hardware. It is also important to schedule installation to minimize the impact on the productivity of the work group.

Phase 3. User training should occur in three steps, preparatory, novice and ongoing.

Phase 4. Maintaining the LAN includes daily maintenance of the peripherals, system backup, user installation, updating software and problem troubleshooting. Most of the maintenance activities can be planned ahead except for problem troubleshooting. Quick problem isolation depends on good network documentation, knowledgeable usage of network diagnostics and common sense.

TITLE: Workload Forecasting for MPE Environments: Methodologies, Techniques, and Tools for the Systems Manager

AUTHOR: Jim Morris

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 2086

TITLE: Architectural Overview and Implementation Methodology of LaserRx

AUTHOR: Ray Ventura

FINAL PAPER WAS NOT AVAILABLE AT TIME OF PRINTING

PAPER NO. 2087