WORKSTATIONS IN A LOCAL AREA NETWORK ENVIRONMENT

S. Yalamanchili R. P. Rao
J. K. Aggarwal M. Malek

TR-82-7 November 1982

Department of Electrical Engineering
The University of Texas
Austin, Texas 78712

*This research was supported in part by a grant from IBM Corporation.

Abstract

Centralized timesharing systems of the 70's are evolving into networks of dedicated workstations and shared resources in the 80's. This trend has come about as a result of the merging of the technologies of local area networks and personal computers. The factors influencing this evolution and the resulting impact on the hardware and software organizations of workstations are described. Guidelines for the evaluation and comparison of workstations are developed. Finally the Appendix includes descriptions of a number of state of the art workstations.

Contents

1. Introduction

The evolution of computer systems has been governed by the need to provide cost effective use of computing resources to individual users. Timesharing and multiprogramming are examples of two concepts that were a direct outcome of this need. As technology has driven the cost of hardware down, it has become possible to provide dedicated computing power to users in the form of personal computers at a cost comparable to that of a terminal connected to a large mainframe - the computer per desk model. However advantages of timesharing systems, such as resource and information sharing are lost when each user is provided with a stand alone system. Meanwhile local network technology has advanced to a state where high speed communication between computers and devices is possible with reasonable response times. As a result the two technologies of personal computers and local networks are merging to provide local networks of dedicated personal computers, called workstations, that possess advantages of personal computers and local networks.

The workstation is intended to satisfy most of the computing needs of a user. The local network provides efficient sharing of expensive resources and high speed communication between users. The development of workstations and their integration into local networks presents organizational and technical challenges. Research efforts to date have succeeded in developing networks of workstations with hardware and software organizations to support a variety of environments, e.g., buisness, scientific etc..

This report integrates information obtained from a study of state of the art workstations. The next section presents the motivating factors behind the development of workstations. The advantages of these systems as well as some of the problems they present are discussed. In the following section the concept of a workstation is refined. The various levels of its organization are identified and each level is then discussed individually. The final section identifies several criteria that may be used in evaluating and comparing workstations. These serve as guidelines in selecting a workstation for a given application. The Appendix contains the descriptions of several state of the art workstations.

2. Local Area Network Based Computing

The design and development of computing systems proceeds along paths dictated by tradeoffs in factors such as technology, cost and performance. Advances in the past decade have combined to produce changing relationships between the above and other design factors. As a result the centralized timesharing systems of the 70's have evolved into network based distributed systems in the 80's. In this section factors that influenced the integration of the technologies of local networks and personal computers are presented. The inherent advantages of these systems as well as some of the organizational and technical problems they present are discussed [BASKET,FAHL,NEWELL].

2.1 Perspective

The pattern of computer usage today entails a number of CPU cycles in interactive tasks such as editing, debugging etc.. This coupled with the fact that CPU cycles are no longer a scarce resource suggests offloading of the common functions to local units. This distribution of computing power to local units has been made possible by the drop in hardware costs. Thus users may be provided with stand alone systems cost effectively.

This availability of localized computing power is important for several reasons. In interactive applications such as software development and use of CAD/CAM tools, the productivity of the user depends greatly on the response time of the system. In timesharing systems response times are unpredictable since they are a function of the load on the system. As the load on the system grows the response times drop off rapidly. The dedicated nature of workstations provides much better response times. In addition the number of CPU cycles provided by a workstation normally exceeds those provided in a timesharing environment. This is especially attractive for compute bound applications such as those involving simulations. In addition certain real time programming applications such as process control or the control of high speed peripheral devices place constraints on response times that cannot be met by timesharing systems. These systems cater to the needs of a large audience, having to provide "average" performance to everyone at the expense of individuals with specialized needs. The presence of localized computing power makes it possible to satisfy users with special needs.

This migration to networks of workstations should not compromise the inherent advantages of centralized timesharing systems. These are coherency among users enabling the sharing of information, efficient communication between users and the sharing of expensive peripherals, which are made available to all users by the central system. Local networks possessing sufficient bandwidth are available which allow users to retain these features in distributed systems. They may also provide for remote links to other networks and thus access to other computing facilities.

The overall design rationale is for timesharing systems of the 80's to exploit differing economies of scale. Share those aspects of a computing system which are not cost effective to distribute. For example a decade ago the CPU was a prime resource to which access had to be controlled. Every user could not be supplied with a dedicated CPU cost effectively. With the advent of microprocessors that is no longer true. However devices such as printers, plotters etc., still remain too expensive to provide each user with and therefore must be shared.

2.2 Motivation

A number of benefits accrue from the networking approach. Most of them are derived from the distributed nature of network based systems. The advantages of these systems serve as strong motivating factors for their development.

In acquiring a computing facility, a major consideration is the outlook for future expansion as the organization and their needs grow. These networks possess a high degree of modularity. As a result incremental expansion is possible through the addition of nodes or gateways to other networks. Integration of nodes into the network is possible locally without necessitating changes at other points on the network. Thus smooth system growth is possible without disrupting network activity.

An important factor is the amount of system downtime that a user experiences. The downtime experienced due to system maintenance is eased since nodes may be replaced or removed without bringing the entire system to a halt. In addition each user has available a constant amount of computing power at anytime independent of the system load. A high degree of fault tolerance is possible. The system may be configured such that failures in individual nodes do not disrupt the system and in fact nodes may be programmed to diagnose each other. In the event of a failure it may be possible for the user to proceed with his work on another workstation. These factors provide the user with high system availability thus increasing his productivity.

The distributed nature of these systems also provides for a wide range in performance. Individual nodes may be upgraded as newer technology becomes available. Special purpose devices may be added as additional nodes and will be available to all the users. In timesharing systems, increase in performance is achieved at a considerable expense whether it is in the form of upgrading an existing system or installing an additional one. For about the same price as that of a large mainframe it is possible to provide each user with a workstation. As a result of their inherent modularity, increases in performance are possible at a lesser cost. Therefore the cost/performance ratio of workstations is usually superior.

2.3 Problems

The advantages presented in the previous section must be weighed against problems introduced due to increased system complexity.

For example though a facility may be available, failure of a link in the network may render it inaccessible from a portion, or all of the network. The failure of a workstation itself may result in the work of a user at that station becoming inaccessible until the cause of the failure can be isolated and rectified. On an overall system basis automatic backup becomes more complicated since files and data are distributed around the network. In addition, due to the proliferation of information around the network, security begins to become a major problem. System maintenance must span the network which may reside over a complete building or several buildings, resulting in increased bookeeping and cost. At the level of the individual users there may be some loss of coherency. Users may be utilizing workstations that are architecturally different. As a result files and data of a user on one workstation may be incompatible with other workstations, thus hindering information sharing. This may lead to several users developing software for the same application. Thus it may be necessary to enforce coherency at some level, e.g., by providing identical workstations or standardizing the high level languages used.

Overall, though they present new problems that need to be solved, networks of workstations and their attendant resource units provide a very flexible base that can be tuned to a wide variety of environments. Table 1 briefly summarizes the advantages and disadvantages of employing network based systems of workstations.

Advantages	Disadvantages
1. Improved respones times	1. Increased system complexity
2. Smooth system growth	2. Complicated file backup procedures
3. High system availability	
4. Increased fault tolerance	3. Security
5. Wide performance range	4. Maintenenace
6. Superior cost/performance ratio	5. Files coherency problem
7. Environment more easily adapted to users needs	6. Need for software compatibility

Table 1. Advantages and disadvantages of employing network based systems of workstations and shared resources.

3. Workstation Organization

The previous section discussed the role of the workstation at a "functional" level. A network wide view was adopted to illustrate the utility of these systems and factors motivating their development. Specifics of the workstations themselves were omitted. In this section the organization of the workstation is discussed. A view of the workstation comprising of several levels of organization is presented. Each level in turn is discussed in more detail.

3.1 What is a Workstation ?

There are a large number of data processing products on the market spanning a wide range in peformance and capability. There is some confusion as to what a workstation is. Due to the increasingly important role workstations are seen to be playing in a number of applications it becomes necessary to identify features that distinguish them from other products. Intuitively a workstation is a facility (a station) that aids the user in performing his work.

Therefore in order to make the above distinction we concentrate on the computing environment that a workstation provides the user. An example of what the user may see at his workstation is shown in Fig. 1. Based on this notion of a workstation environment we identify three properties that it is felt a workstation should possess.

1. Local computing power
2. Access to all facilities necessary for the working environment of the user
3. Inter-user communication

In our opinion any hardware module that satisfies the above three properties may be termed a workstation. The implications of each are discussed below.

The first property is the primary distinguishing factor between network based systems of workstations and centralized timesharing systems. Local computing power is the availability of a certain amount of dedicated machine cycles which are not shared with any other user. At the very minimum this necessitates a processor and a limited amount of memory.

The second property states that if resources such as printers and plotters are required, the workstation should

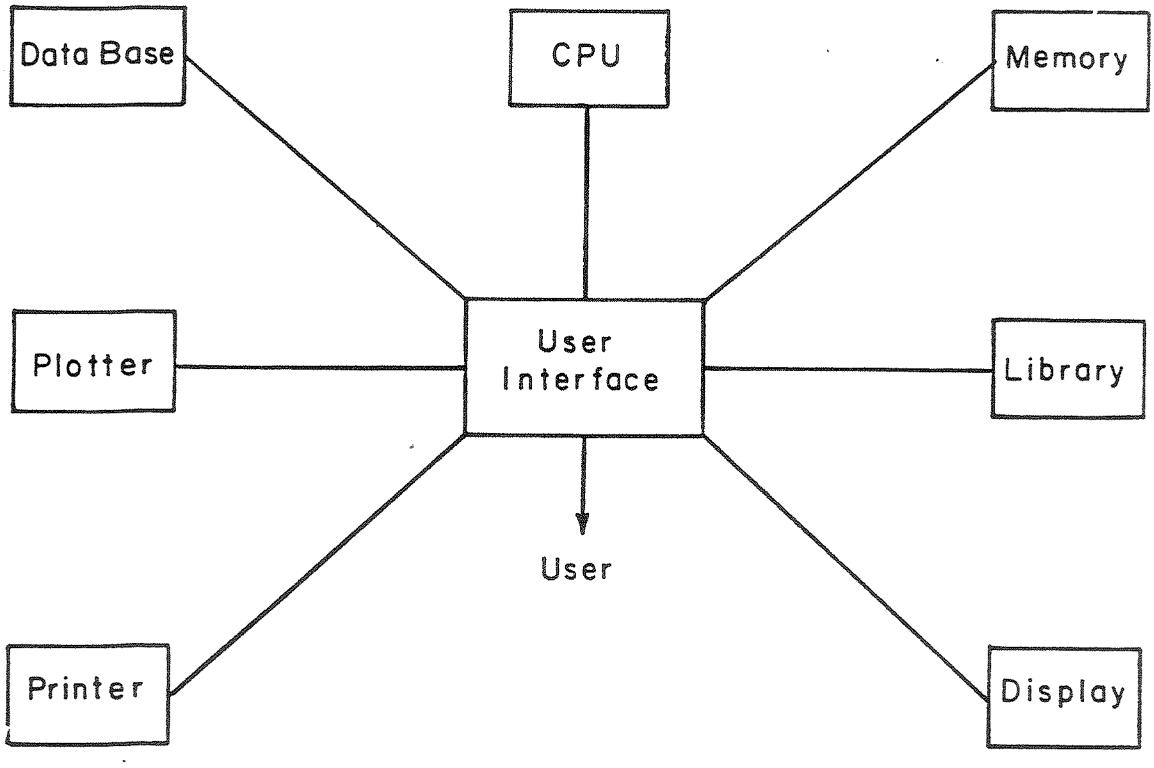

Figure 1. An example user environment.

provide a means for access to them, in most cases through a local network. The third property i.e., inter-user communication, also implies the presence of some local network. The implication of the last two properties is very important. In essence they state that a workstation is more than just a compact locally available resource. It should provide a complete working environment. All the facilities required by the user for his work should be provided to him by the workstation or through the network. These considerations suggest that any system that provides a user with a working environment can be termed a workstation.

Having identified properties characteristic of workstations we now focus on their internal organization. At the very lowest level is the hardware that physically executes the workstation functions. This is comprised of a dedicated CPU supported by local primary memory, possibly secondary memory, and I/O devices for communication with the user and the environment. The latter usually includes a bitmapped graphics display device. Though an expensive part of the workstation the immense human factor advantage it provides in many applications justifies its inclusion. The details of this level are generally transparent to the user.

Above this level is the software level. This includes the local operating system and utilities for file management, electronic mail etc.. The hardware and the software combined give the workstation its functional nature. The functional level is typically what the user perceives. The characteristics of this level gives the user an idea of the capability of the workstation, and therefore allows him to gauge how well it will fulfill his needs. The highest level is the user interface. This is the level at which the user interacts with the workstation. This level is very important since it determines such qualitative metrics as ease of use and flexibility. As personal computing rapidly expands into primarily non-technical areas, how "close" the workstation is to the user gains importance.

The levels of organization are shown schematically in Fig. 2. The following sections discuss each level. Since the hardware and software organizations determine the characteristics of the functional level, this level is omitted from further discussion.

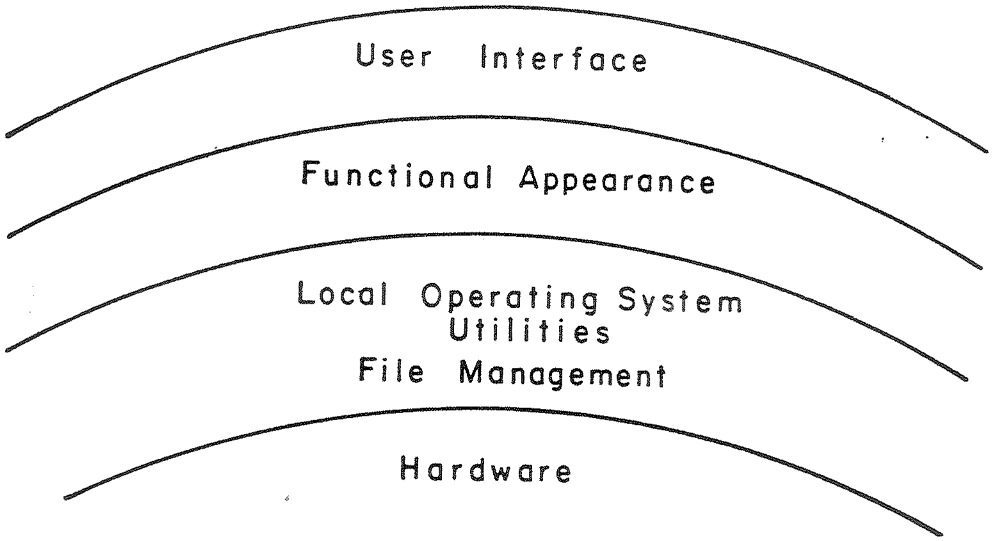

Figure 2. The various levels of organization in a workstation.

3.2 Hardware Organization

The hardware is typically comprised of the following four subsystems,

1. Processor Subsystem
2. Memory Subsystem
3. Graphics Subsystem
4. Input/Output Subsystem

Workstations differ mainly in the development of and relationships between, the individual subsystems. In the following each of these subsystems is discussed.

3.2.1 Processor Subsystem

The processor subsystem is most commonly based on a 16 bit microprocessor. Currently available microprocessors include the MC68000, Z8000, and Intel's 8086 and 8088. These processors provide computing speeds of the order of 1-8 MIPS.

The instruction set of the processor and the flexibility it provides is an important issue. Flexibility may be afforded through user programmable microstore. Such a capability is desirable for several reasons [NEWELL]. Machines specialized for various languages and applications can be created by modifying the instruction set. As needs change, new instructions can be created or old ones modified extending the useful life of the instruction set. It enables the user to pursue his own software and language development. For example it is possible to create special instructions for network communications and file management. Instructions or small commonly used routines can be placed in microcode optimizing special purpose functions. Such a feature is useful if, for instance, the processor is to manage high speed devices or in the real time programming of special purpose devices where strict timing requirements must be met. Programmable microstore may also impact the overall design of the workstation. Many of the control operations of peripheral devices may be implemented in microcode in the workstation processor. This significantly reduces the complexity of the device controllers thus allowing for a more compact design. However, this approach does tend to customize the processor to a given application. In addition very often the user does not want to be burdened with the task of microprogramming. The use of a general purpose processor with high level language support is desirable when the environment covers a wider range of applications and is less predictable.

3.2.2 Memory Subsystem

The memory subsystem may provide both primary memory and local secondary memory. The limiting factors on memory sizes are cost and physical volume. Typical primary memory sizes range from 128 kbytes to 2 Mbytes and for secondary storage from 5 Mbytes to 500 Mbytes. Local secondary memory may be abandoned in favor of the use of local file servers that are distributed throughout the network. The advantage of this approach is that the user is not necessarily tied down to one workstation. However the presence of secondary storage provides memory for storing more software and data, e.g., character fonts, integral to a workstation. Physical and economic limitations on memory size mandate virtual memory capability for many applications, especially in a scientific environment. Many companies are currently developing a single chip memory management unit to handle the paging and address translation mechanisms. This may dramatically change the design philosophy of providing users with large address spaces. Both design approaches, ie. with or without virtual storage are feasible and depend on user requirements.

3.2.3 Graphics Subsystem

This system was a very innovative addition to the workstation.

Graphics capability is commonly provided through a bit mapped raster display. High resolution displays typically consist of 1024X1024 bits and medium resolution displays 512X512 bits. Graphics devices place high demand on the CPU and the internal communication devices of the workstation due the large amounts of data that have to be transferred to and from display memory. Therefore a major concern are mechanisms for fast and efficient means for manipulating the display memory and controlling the refresh cycles. One method is to implement display memory as a dual port memory with one port dedicated to display and the other to update operations, relieving the CPU of the task of coordinating the refresh operation. In addition special hardware may be provided for moving areas around the display, and to and from primary memory. This would enable the processor to perform raster operations in primary memory and use the dedicated hardware to perform the transfer. This approach may be supported by providing optimized microcode routines for display functions.

An additional feature that may be desirable is the capability for modular expansion or the adaption of the graphics subsystem to color or greyscale monitors. This may be required in some applications, e.g., VLSI layout, design automation etc.. This would necessitate a degree of flexibility in the graphics subsystem. It should be possible to provide timing signals for different types of monitors. In addition the mapping of the bitmap onto the display changes over monitors (binary to grayscale to color) and has to be accommodated. One possible organization of this subsystem is the following [BECH82]. A dual port frame buffer for the bitmap with one port dedicated to refresh and being connected to a programmble device specialized for formatting functions. A separate programmable module generates the basic timing signals. Such a modular construction allows use with images ranging from binary to color. Initial investment however is higher than for a system that requires the processing of only binary images. Additional tradeoffs have to be made concerning the hardware vs software implementation of display manipulation functions.

3.2.4 I/O Subsystem

The I/O subsystem should address three aspects. The interface to the network, the interconnection of devices integral to the workstation and the interconnection of peripheral devices to the workstation. Additional performance criteria also determine the organization of this subsystem. For example, minimum acceptable response times when communicating over the network and/or frequency of display operations may necessitate customized channels between some devices, e.g., DMA channel between memory and the network interface. More specifically, transparent access of files across the network dictates that file transfers should proceed at maximum possible bandwidth - primary memory bandwidth. Thus the network should have full bandwidth access to primary memory. The graphics subsystem in turn may be relegated to a physically separate communication device to increase the communication bandwidth available to the other devices. An additional consideration is interfacing to devices providing audio and video I/O, e.g., speech synthesizer, video recorder etc.. These devices place additional demands on the available system bandwidth. Thus given the wide range in peripherals, it may be necessary to provide separate communication devices for high speed devices and low speed devices.

This section has described the various subsystems that typically comprise a workstation with a view towards identifying the choices that had to be made in designing them. Similar tradeoffs have to be applied to the software organization which is discussed in the following section.

3.3 Software Organization

The software organization in conjunction with the hardware determines the functional characteristics of the workstation. The control at this level resides with the local workstation operating system. This local operating system tends to differ from operating systems of centralized timesharing systems. This is a result of the single user environment [NEWELL]. Memory protection and scheduling tend to decrease in importance while network communication and I/O (interacting with the user) increase in importance. However even in this single user environment, demands for increased speed and efficiency are leading to local operating systems that support multiple concurrent processes thus requiring memory protection schemes. This again mandates that support be available for fast context switches and the saving of state information of suspended programs. On a network wide basis the global operating system may be either centralized or distributed. The design of these network operating systems raises new questions. One major question is whether traditional operating systems whose semantics are based on addressable memory can be extended to network based systems. In particular, one research effort has focussed on developing a network operating system whose semantics is based on communication primitives [WARD]. Traditionally passive structures such as data files and program files are implemented as active processes. The primitives focus on the operation of communication "streams" between active processes. This view supports the 'abstraction of function from implementation' approach. Thus actual implementation of the active processes may vary providing a wide range in performance.

Virtual memory management is another important function of the local operating system. The system should maintain the tables that provide the information for the mapping of logical addresses to physical addresses. In addition, memory pages have to be swapped into and out of main memory from secondary memory. If no local secondary storage is available, paging service must be provided from a local file server. If a network wide virtual memory system is adopted, the global operating system must maintain tables that provide information for the mapping of files across the network into a users local logical address space. In such cases network congestion may result in users experiencing poor response times. Because of the demands on response times, various techniques are used to speed up the process.

The design and management of the network file system becomes particularly important. Large centralized file facilities have advantages of being easier to maintain and providing sharing of files among users while decreasing storage requirements at the workstations. Distributed file systems provide advantages in fault tolerance and better response times. However protection and transparent access of files throughout the network becomes increasingly difficult. Considering the size of a network based system and the large numbers of files and users, flexibility in locating and accessing files is important. Tree structured directories is one organization that is commonly used. Users may attach themselves to any node on the tree and access files specified by the corresponding subtree. Specification of a file requires specification of a path on the tree to the required directory. Nodes (directories) in the tree are specified by symbolic names for ease of use. The adoption of a particular file system is best done based on the environment, since this would indicate the manner and frequency with which files would be accessed around the network.

The programming environment tends to vary to a great degree across application domains. In an office environment text formatters, screen oriented editors and graphical illustrations attract a great deal of attention. Electronic mail, personal communication facilities to handle interoffice memos, business software for market analysis, short term and long term scheduling etc. are but a few of the utilities that would be necessary in this scenario. In a scientific environment, one would more commonly come across support for several high level languages, symbolic debuggers, runtime libraries etc.. One may assume that given a modularly expandable hardware basis, it would be possible to develop software tools for most applications.

3.4 User Interface

The highest level of organization in the workstation presents to the user a "virtual machine" - the user interface. The dedicated nature of the workstation can be used to provide the user with an extremely versatile and sophisticated interface, which opens up new possibilities in bringing the machine closer to the user. This factor becomes increasingly important as workstations penetrate markets such as office automation, where novice users with no technical background may be required to use the workstation. The essential feature of the workstation that makes the user interface important is that it is primarily an interactive system.

When designing an interactive system several important features must be kept in mind. The system should project a natural and uncomplicated image of the computing resources. An inexperienced user should be able to learn how to use them without difficulty. Therefore there are two main points to consider, communication from the workstation to the user and vice versa.

In the case of the former, communication from the system to the user is no longer limited to the visual display of messages, but may also include audio and graphics. The former may be in the form of prerecorded messages serving as reminders or instructions. The latter may be useful in portraying physical devices such as folders, mail baskets, etc., on a screen, so that the user may select one by a pointing device for further examination. Extensive use is made of on-line documentation and tutorials. One very desirable feature is the use of dynamic heirarchical command menus . At each stage of system use the user is presented with a set of possible commands. Selection of one command will result in the presentation of the next set of commands and so on until the required function is accomplished. A good survey of menus as well as tradeoffs of using dynamic versus static menus are given in [PRICE].

A variety of media are available for user communication to the system. Speech understanding systems with small vocabularies have been successfully developed and their integration into workstations is seen to be not far off. Typed commands from a keyboard still form the largest percentage of the communication, and hence the flexibility of the command structure is important. Pointing devices such as the mouse are very useful in marking sections on a display screen that are to be operated upon, e.g., in text editing . In addition touch sensitive screens, tablets for input of graphic data [OHLSON] and multifunction keys are used.

The system should respond quickly and meaningfully to user commands. It is expected by the user that the response time is a function of the complexity of the task. However it may not be acceptable for functions that are used very often to have very long response times, irrespective of their actual complexity. It is important also that all actions are initiated and controlled by the user. This desire for control of the computing environment appears to increase with the experience of the user.

The command sructure should be flexible and forgiving of errors. Specifically the system should protect the user against making costly errors. For example a file may be accidently overwritten. Mechanisms should be provided within the system to avoid such errors. This is sometimes referred to as "undo" or "redo" capability. This can possibly be achieved by preserving the state of the system at various points within the user's current session. If at some point the user has realized he has committed an error, he should be able to restore the system to its latest saved state.

Finally it is desirable that the user interface presents a uniform image across all domains. The command structure should therefore be consistent and robust. This makes it easier to learn and convenient to use and unecessary to get reacquainted with for each new application.

4. Evaluating Workstations

Given the wide range in capabilities in available products it is often difficult to determine which workstation best satisfies a user's needs. This section presents guidelines for evaluating a workstation with respect to a specific application. Several criteria are discussed and some benchmarks proposed to aid in a comparative evaluation of workstations.

4.1 Criteria

The user's interaction with the system is at the functional level. Therefore he is more interested in how well the hardware and software are integrated into the desired functionality, rather than their specific characteristics considered individually. When evaluating a workstation the following major criteria should be considered: functionality, user interface, reliability, price and flexibility [RICH].

Functionality- The functional capabilities of the workstation should match the needs of the particular application. In this respect, it should be verified that application requirements for desired functions such as text editing, electronic mail, etc., are provided by the workstation. Then a comparison can be made between the various workstations that provide these functions, with respect to the power and quality of their implementations.

User interface- This addresses the issue of how easy the system is to learn to use and of how convenient it is to use.

This consideration may perhaps be the most important. For example it is more desirable to have a system of average functional utility but high user appeal, and hence used more often than a functionally powerful system that is complicated and difficult to use. The beginner should be able to start using the system immediately in a limited manner, with the aid of structured menus and on-line tutorials. At the same time however, the expert user should have the means for accessing the full power of the system. The appeal of the system to the user depends on such factors as response time, protection of the user from errors, power of communication between system and user and the physical appeal of the product.

Reliability - In attempting to determine how reliable a workstation is, it is desirable to include characteristics of the network. The workstation may not be of any use if a required resource is not accessible due to a network failure. In the event of the failure of a workstation it should be possible to continue working on another workstation that is attached to the network. This would require that multiple copies of all necessary files be maintained. In addition the failure of a part of the network should not prevent the rest of the network from working correctly, albeit in a reduced manner. If the network has any critical components, their susceptibility to failure must be evaluated.

Price- It is difficult to make a precise comparison of workstations with regard to their price because of the differences in the services that they offer. However, to make a rough comparison the total cost of the workstation must be known. This includes the initial hardware and software costs to make the system operational, and the costs of maintaining the system in the future. It is also important to consider the incremental costs in expanding the system at some later date.

Flexibility- It is expected that the needs of the user will change and grow with time. Therefore it is important that the system be designed so that it can evolve and grow with the needs of the user. It should also be able to be upgraded to keep up with changes in technology. New software and peripherals should be able to be added as they become available.

4.2 Benchmarks

While the hardware structure and operation rates are the principal performance determinants, the overall system organization is important. Besides the basic functions which are required for a general purpose workstation there may be special needs. Employing a methodology similar to that employed in evaluating conventional computer systems, benchmark programs may be used to aid in workstation evaluation.

In our opinion the following benchmarks could be used.

1. Numeric Processing
   - a. Basic instruction set execution
   - b. Floating point arithmetic
   - c. FFT (Fast Fourier Transform)
   - d. Sort and merge functions (used in file manipulation)

e. Virtual memory space exchange

2. Non-Numeric Processing

   a. Word processing (Text formatting and retrieval)

   b. Speech recognition and generation

   c. Buisness graphics

   d. Boolean matrix transpose

   e. Code converter (e.g., ASCII to floating point)

3. Memory Management and Security

   a. Database management

   b. File handling

4. Network Capabilities

   a. Network file retrieval from other stations

   b. Electronic mail

   c. Files coherency

5. Understandability

   a. Ease of use and learning

   b. Understanding the application of powerful features

The last benchmark is difficult to measure because of the inherent human factors involved. It is clear that the above focuses mainly on the problem areas for which benchmarks should be defined. There is still a need for precise definitions of problems whose computation on different workstations may be used for effective and true comparison.

5. Concluding Remarks

Network based distributed computing systems are the systems of the future. Workstations are an integral part of such an environment. Their development and use will have a significant impact on the direction of development of future systems.

This report has integrated the results of a study of currently available workstations. The design rationale and motivation behind the evolution of network based systems of workstations and their attendent advantages and disadvantages were discussed. The concept of a workstation was refined and a hierarchical view of the workstation organization was proposed. The considerations involved in the design of each level were then discussed. Finally the problem of evaluating workstations for a given application was introduced. Several criteria and benchmarks were outlined. Further work is required in this area to establish a systematic methodology for workstation evaluation. This is important in view of the expanding role workstations are seen to be playing in buisness, scientific and in an increasing number of new environments.

References

[APOLLO] Apollo Domain Architecture, Apollo Computer Inc., N. Bellerica MA, Feb. 1981.

[BASKET] F. Baskett, A. Bechtolshiem, B. Nowicki and J. Seamons, The SUN workstation : A terminal system for the Stanford University Network, Department of Computer Science, Stanford University, March 1980.

[BECH80] A. Bechtolshiem and F. Baskett, The SUN workstation : A hardware overview, Department of Computer Science, Stanford University, Dec. 1980.

[BECH82] A. Bechtolshiem, F. Baskett and V. Pratt, The SUN workstation architecture, Computer Systems Laboratory, Stanford University, Jan. 1982.

[CLARK] D. W. Clark, B. W. Lampson and K. A. Pier, The memory system for a high performance personal computer, IEEE Trans. on Comp., vol. C-30, no. 10, Oct. 1980.

[FAHL] S. E. Fahlman and S. B. Harbison, The SPICE project, SPICE document S101, Department of Computer Science, Carnegie Mellon University, May 1982.

[HUIE] J. Huie, Workstation clusters expedite local network processing, Electronic Design, pp. 129-134, Feb. 1981.

[LAMP] B. W. Lampson, G. A. McDaniel and S. M. Ornstein, An instruction fetch unit for a high performance personal computer, to appear in IEEE Trans. on

Comp..

[LAMP80] B. W. Lampson and K. A. Pier, A processor for a high performance personal computer, Proc. of the 7th Annual Symp. on Comp. Arch., pp. 146-160, May 1980.

[MCDAN] G. A. McDaniel, The Dorado : A compact high performance personal computer for computer scientists, COMPCOM, pp. 479,Spring 1980.

[NEWELL] A. Newell, S. E. Fahlman, R. F. Sproull and H. D.Wactlar, CMU proposal for personal scientific computing, COMPCOM, pp. 480-483, Spring 1980.

[OHLSON] M. Ohlson, System design considerations for graphics input devices, Computer, pp. 9-18, Nov. 1978.

[PERQ] A Landmark Computer System, Three Rivers Computer Corp., Pittsburgh, PA.

[PERQ82] Product Description, TRC-6, Three Rivers Computer Corp., Pittsburgh, PA.

[PRICE] L. A. Price, Design of command menus for CAD systems, Proc. of the 19th Design Automation Conf., pp. 453-459, July 1982.

[SEYBOL] Selbold Report on Word Processing, vol. 4, no. 5, May 1981.

[TERMAN] C. J. Terman and S. A. Ward, An approach to personal computing, Laboratory for Computer Science, M.I.T.,Dec. 1979.

[RICH] F. K. Richardson, Important criteria in selecting engineering workstations, Proc. of the 19th Design Automation Conf., pp. 440-444, July 1982.

[THACK] C. P. Thacker, E. M. McWright, B. W. Lampson, R. F. Sproull and D. R. Boggs, Alto : A personal computer, CSL-79-11, Xerox Palo Alto Research Center, August 1979.

[WARD] S. A. Ward, Trix : A network oriented operating system, Laboratory for Computer Science, M.I.T., Dec. 1979.

Appendix

Πhis section provides descriptions of workstations currently available. In describing individual workstations, unique features in terms of organization or novel implementations are discussed first. This is followed by a summary of the characteristics of the subsystems that it is comprised of.

A.1 Apollo

The Apollo workstation is designed for use with the Apollo Domain system [APOLLO] based on a ring network topology. The motivating factors in its development were the need to combine the advantages of timesharing with the fast response times afforded by dedicated minicomputers.

The I/O system of the workstation is divided into two parts. The first part is devoted to those devices which are an integral part of the workstation, i.e., the processor, disk subsystem etc.. These devices are connected to a block multiplexor channel. Other user supplied peripherals are connected via a multibus controller. As a result, the memory management system is divided into two parts, one for the CPU and the other for the I/O subsystem - peripherals on the multibus. Fig. A1 shows the hardware organization of the Apollo workstation.

One of the main objectives of the Apollo system was to maximize network responsiveness. This is to make transparent to the user the physical access of files and data across the network. This need dictated that the network and the local secondary storage have full bandwidth access to primary

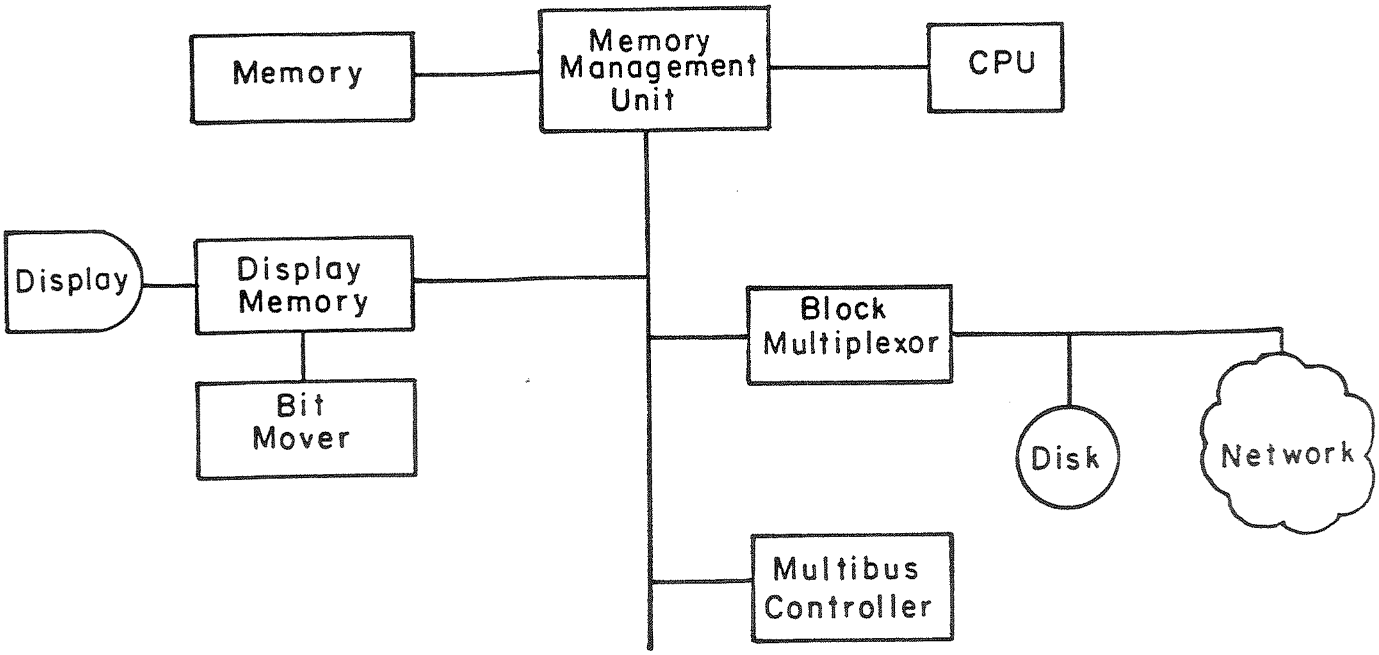

Figure A1. The Apollo workstation.

memory. This is achieved in the Apollo system by having both disk and network accesses pass through the block multiplexor channel, in effect sharing a DMA channel into primary memory. Transfer to and from memory takes place at almost 100% bandwidth.

In designing a processing environment, programs and data files are abstracted into objects. The complete space of the operating system and all objects form a a network wide 96 bit virtual address space. Each object has a unique 64 bit identifier and a 32 bit address space. The individual workstation processes have a 24 bit virtual address space which is translated by the memory management unit into a 22 bit physical address. The memory management system has the necessary support for demand paged I/O. When an object is referenced on a distant workstation for example, the operating system maps that object into the virtual address space of the process but does not initiate any data transfer until the CPU makes a reference to that virtual address. When a reference is made, transfer across the network takes place at full primary memory bandwidth. In a similar manner using the second part of the MMU, I/O devices on the multibus can address themselves into a selected number of pages of the CPU. These mechanisms enable interprocess communication based on mapped primitives which is independent of the location of the processes on the network. It is implemented as demand paged I/O at the page level. Therefore a user actually sees nothing more than a central file facility, all or a part of which is accessible depending upon access privileges. This manner of employing mapped primitives enables the implementation of a very powerful command language called the shell programming language. A shell program can coordinate the execution of multiple programs with serial or parallel relationships.

Currently a new version of the Apollo workstation is being developed. It should maintain all the advantages of the old model combined with a more attractive price in the neighborhood of $10,000.

System Characteristics

Processor Subsystem

- MC 68000 16 bit microprocessor

Memory Subsystem

- 4 Mbyte of primary memory protected with error detecting codes
- Local secondary storage
- 24 bit virtual address space
- Demand paging
- Separate memory management units for peripheral devices and devices integral to the workstations

Graphics Subsystem

- 1024X1024 bitmap with 800X1024 refreshed onto a display and the remainder used as fast scratch storage
- Bit mover for manipulation of blocks within the display and to and from primary memory

I/O Subsystem

- DMA channel between network and primary memory, and disk and primary memory
- Multibus controller for access to devices on the

multibus

Software

- Display manager
- Virtual terminals

A.2 Nu

The Nu system [ΓERMAN,WARD] being designed at the Massachussets Institute of Technology, though possessing similar overall objectives may be contrasted with the Apollo system both in hardware and software organizations. Two main objectives were performance scaling and abstraction of function from implementation, the latter objective in many ways derived from the first. This has a significant impact both on the workstation organization and operating system characteristics.

The Nu system is designed around the Nu bus. This is a 32 bit wide bus. Each bus can support up to 32 separate modules connected to it. This bus is the vehicle for all communication. There are no customized data paths, e.g., memory to display. Every module appears identical to the bus. This organization is a result of the abstraction of function from implementation objective resulting in a type of "tinkertoy principle": every plug fits every socket. This bus is allocated by a distributed arbitration scheme which reduces the latency time over centralized arbitration. Due to its generality the bus represents a hospitable basis for expansion. Scaling of performance by addition of modules may be done, while protecting the investment in other devices. This is due to the uniform interface of every module to the bus. There are some costs associated with this policy. Even the most trivial communication requirements require the same mechanisms as more demanding needs. This is accepted as the tradeoff for organizational simplicity and flexibility. The bus is not expected to be a bottleneck since larger data path widths and increased bandwidth my be afforded by using parallel buses. In order to investigate fully the potential benefits of the Nu system, a 32 bit microprogrammable processor, the Rho processor, is being used to prototype future applications. A novel feature is the inclusion of two Nu bus interfaces in the organization of the processor. Thus applications requiring excessive communication bandwidth, e.g., video graphics, may be relegated to a separate bus and interfaced to the main bus through the Rho processor. A schematic of such an organization is shown in Fig. A2. With this basic building block structure and the Rho processor, it is possible to envisage a variety of configurations yielding a large range of performance characteristics.

The operating system for the Nu [WARD] system reflects the same goals as the hardware organization. The operating system semantics are based on communication primitives rather

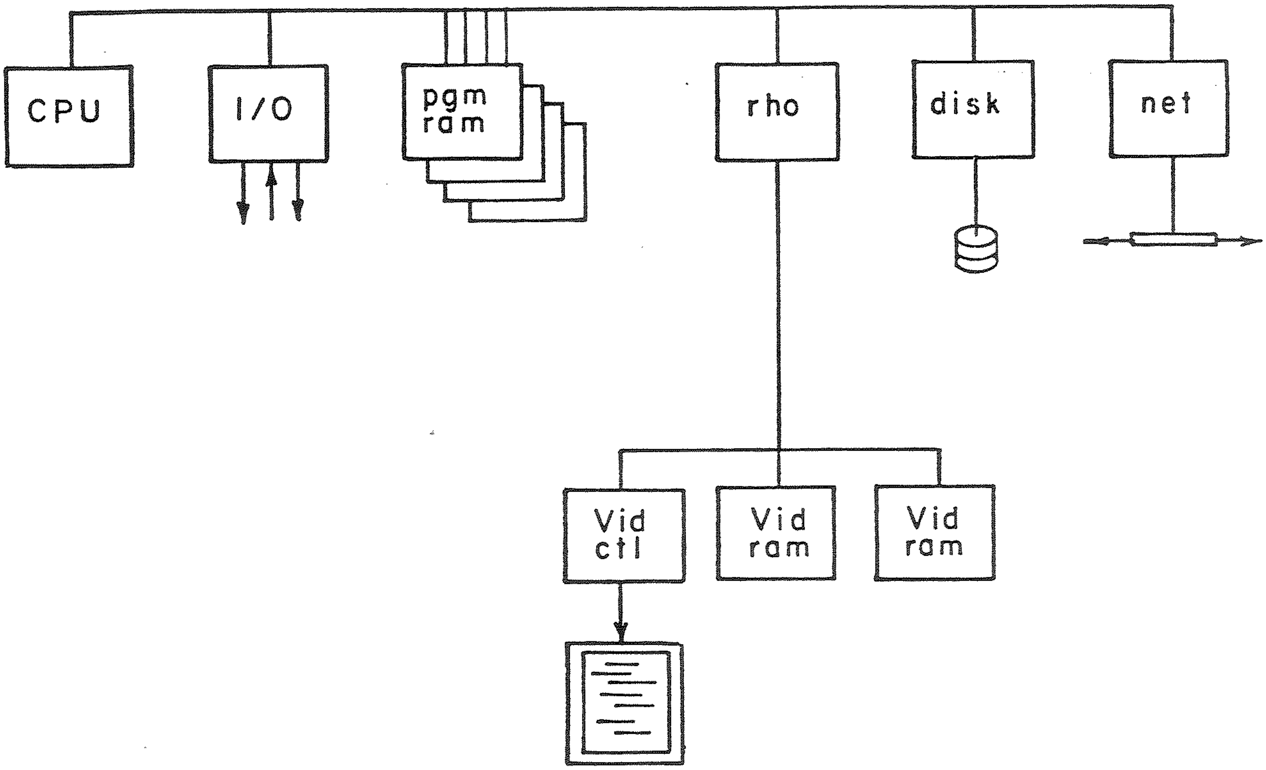

Figure A2. A system configuration based on the Nu bus.

than traditional addressable memory. Passive objects such as files and programs are implemented as active processes. Two processes are connected by a communication stream. Communication streams to a process can be read or written. Thus on a read the process may be implemented as a passive file or a hardware device that computes the required values on request. Any implementation is acceptable as long as its I/O requirements to the streams that it is connected to are satisfied. This approach is in tune with the abstraction of function from implementation objective. The environment of a workstation process consists of the set of communication sstreams it is connected to and hence can be tailored by other processes which may pass access rights to a part or whole of their environment.

Such an approach based on communication primitives does introduce some problems. Implementation of program and data files as active processes creates problems during system crashes when all active files are lost. The implementation should therefore include reliability considerations. The management of streams to and from removable volumes also presents some problems. Stream communication is a voluntary act and therefore consideration of message buffering and control is necessary. In conclusion it is apparent that this operating system is intended to impose very few constraints at the user level while potentially supporting the objectives of performance scaling through abstraction of function from implementation.

System Characteristics

Processor Subsystem

- MC 68000 or Z8001 16 bit microprocessor
- Timer
- Serial and parallel I/O interfaces

Memory Subsystem

- Primary memory available in 128 Kbyte modules
- 10 - 45 Mbytes of secondary memory
- 24 bit virtual address space
- DMA between primary and secondary memory

Graphics Subsystem

- Microprogrammable synch generator which may be programmed to support grayscale and color monitors
- Video formatter to format bitmap for display and can be adpated to support grayscale and color monitors
- 1024X1024 bitmap
- Mouse pointing device

I/O Subsystem

- 32 bit wide Nu bus (excluding 16 control lines)
- 32 bit Rho processor for protyping future

applications

- Interface to a 8 1bs network
- All modules appear identical to the Nu bus

Software

- Operating system based on communication primitives
- Files and programs implemented as active processes
- Software tuned to maximize abstraction of function from implementation

A.3 Perq

Perq [PERQ,PERQ82] is a workstation marketed by Three Rivers Computer Corporation of Pittsburgh, Pennsylvania. The local network employed to interconnect the workstations is the 10Mbs Ethernet baseband network. Upto 1024 Perq workatations can be connected over a 2500 ft cable.

The processor subsystem is built around a microprogrammable 16 bit minicomputer. The hardware organization of the Perq workstation is shown in Fig. A3. The native instruction set is the Q code byte sequences that a compiler generates for an ideal pascal machine. The processor is capable of executing approximately 1 million of these Q code instructions/sec. The instruction set is stored in a 24 kbyte writable control store. This store supported by special purpose hardware for emulating byte coded instruction sets gives users the ability to pursue their own language development. The microprogrammed CPU also performs many of the bookkeeping operations of attatched devices integral to the workstation. Using the microcode in this manner reduces the complexity of device controllers significantly. In addition commonly used display functions are optimized in microcode. The video controller itself is

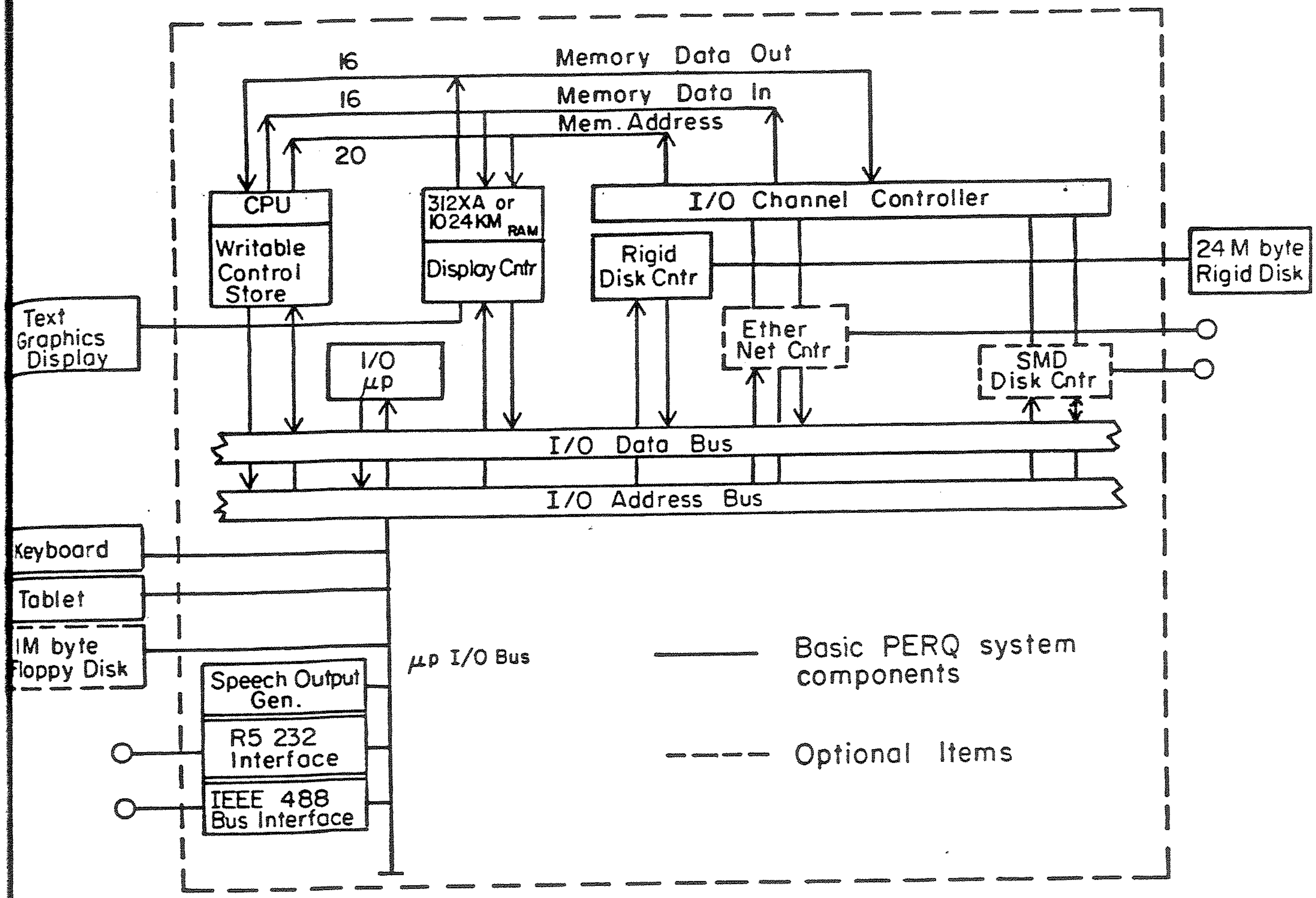

Figure A3. The Perq workstation.

integrated with the memory controller further contributing to a more efficient and compact design. Meanwhile to support the high bandwidth of the various devices, especially the display, the memory system is designed with a 200 Mbit aggregate bandwidth and 50% of the bandwidth of the internal buses is dedicated to the display, i.e., every alternate cycle, with the remaining 50% of the bandwidth shared among the other devices.

The I/O subsystem provides multiple data paths and two separate I/O controllers. One I/O controller dedicated to high speed channels multiplexes four high speed devices which are the rigid disk, the network and two optional controllers. A second, low speed controller consists of a Z-80 8 bit microprocessor for interfacing to laboratory devices and perpherals possessing less stringent bandwidth requirements. Among the devices communicating through this controller is a speech synthesizer. Prerecorded data is stored on the system and used for audible signalling, voice response and other speech applications. The interface to the IEEE 488 standard general purpose instrumentation bus provides a simple compatible way of interfacing to a wide range of medium speed peripherals such as laboratory facilities. A three bus design is used for communication between CPU, memory and high speed controller, and represents the communication system for devices integral to the node. Communication between these devices and the I/O controllers is supported by a separate I/O bus. A very useful feature of the Perq system is the inclusion of diagnostic software. When the system is booted, overall system operability is checked and faults, if any, are isolated to replacable units.

Though the Perq workstation has been successful so far, its limited address space and its relatively high price ($20,000) may soon affect its marketability.

System Characteristics

Processor Subsystem

- 16 bit microprogrammed minicomputer
- 24 kbyte writable control store for language development and functional optimization

Memory System

- 512 kbytes primary memory expandable to 1 Mbyte
- 24 Mbyte secondary memory
- 1 Mbyte floppy disk available as an option
- Virtual memory system featuring a 32 bit segmented address space with address translation supported by hardware and microcode

Graphics Subsystem

- 768 X 1024 bitmap
- Display functions optimized in microcode
- Pen style pointing device

I/O Subsystem

- High speed I/O controller multiplexing 4 high speed channels
- Z-80 based low speed controller for interfacing low speed devices
- Speech synthesizer for audible signalling and

speech applications

- IEEE 488 bus interface

Software

- Perq operating system which supports a single process environment
- Display manager
- Screen oriented editor
- Fortran and Pascal high level languages
- Version 7 of UNIX

Options

- Disk controllers with device capacities of 40 - 300 Mbytes upto a maximum of 4 drives
- Daisy wheel printer

A.4 SUN

The SUN workstation [BASKET,BECH80,BECH82] was developed for use on the campus wide 3 Mbs Ethernet network at Stanford University. The workstations were intended to provide a hardware base for research in VLSI design automation, advanced text processing etc. rather than as a research problem itself. Consequently goals included a low cost system that could be built from off the shelf LSI components to satisfy immediate needs. The low cost was to be supplemented by a modular scalable design that would make it cost effective to consider versions of the basic workstation as interfaces to clusters of remote terminals, printer server, file servers etc. Its utility is intended to run beyond providing localized computing power but without being economically prohibitive.

The basic SUN workstation does not provide any local secondary memory. This is provided through a local file server which may itself be a modified SUN workstation. Such a configuration is shown in Fig. A4. A cluster of SUN stations are connected via a remote Ethernet. The file server provides a bridge to the main Ethernet, local cache for commonly used files and provides swapping and paging

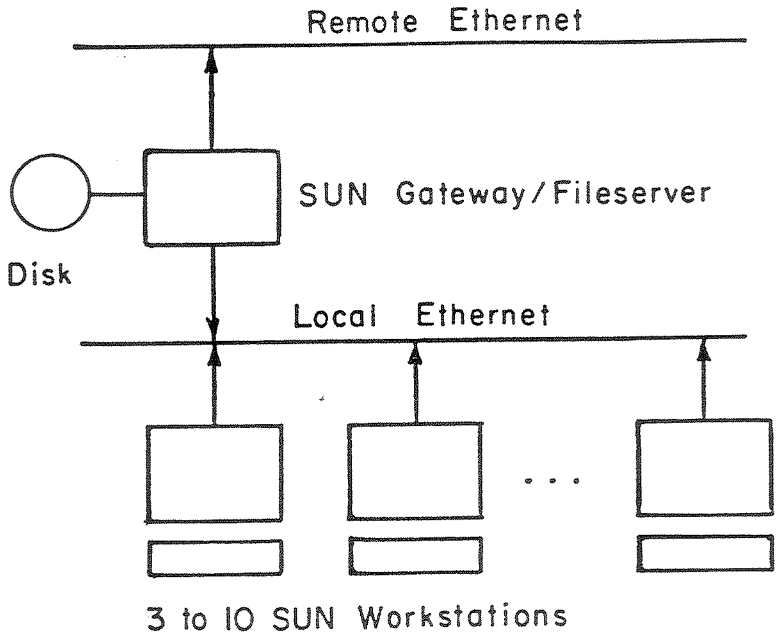

Figure A4. A configuration of SUN workstations.

services, isolating traffic from the main ethernet. This approach is intended to provide for efficient sharing of files. The file system though appearing logically centralized is physically distributed among the various file servers.

The I/O subsystem provides for communication between workstation integral units and peripherals through the multibus which is being standardized as the IEEE 796 bus. This choice was motivated by the decision to build the workstation using off the shelf components. In addition, being accepted as an industry standard has made it possible to obtain a wide variety of devices compatible to the multibus. The network interface is configured as a separate module on the bus. This module implements the data link layer and physical layer functions of the ethernet.

The SUN's basic design was intended to make it useful for a wide variety of functions. Some configurations are shown in Fig. A5.

A commercial version of SUN including 1 Mbyte of primary storage and 85 Mbytes of disk storage is currently being marketed for about $20,000.

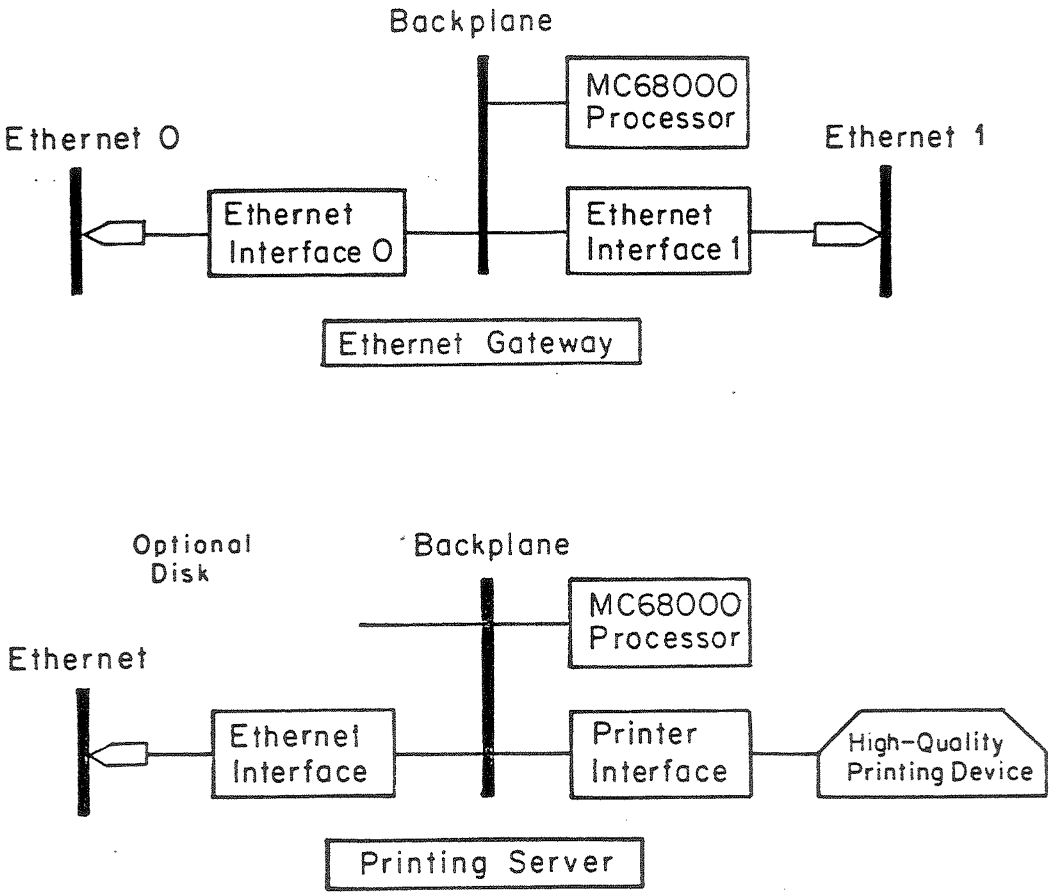

Figure A5. Two examples of SUN workstations configured for different applications.

System Characteristics

Processor Subsystem

- MC 68000 16 bit microprocessor
- 2 high speed serial communication channels
- 16 bit parallel input port
- Five 16 bit timers for automatic control in remote applications
- Slave port for privileged communication between workstations

Memory Subsystem

- 256 kbyte primary memory expandable to 1 Mbyte
- Basic unit has no secondary memory
- 21 bit virtual address space
- Demand paging

Graphics Subsystem

- 1024 X 1024 bitmap of which 800 X 1024 is refreshed to the display and the remainder is used as fast local scratch storage
- Graphics controller. May be programmed to support grayscale or color monitors and both interlaced and noninterlaced refresh format for medium or high resolution displays

- 1 X 16 bit update harware primitive
- software defined variable size character fonts
- Upto 8 frame buffers/controller

I/O Subsystem

- Modules interconnected by the standard IEEE 796 Multibus

Software

- Workstation operating system supports upto 16 simultaneous contexts, expandable in software
- Memory protection through the use of base and limit registers

A.5 Alto

The Alto [THACK,SEYBOL] is a workstation designed early in 1973 as an experiment in personal computing by Xerox Corp.. Since then it has undergone a number of changes and improvements. The details provided in this section are with reference to the latest model.

The Alto functions as 16 microcoded tasks that share the processor and memory subsystems. These tasks have fixed priorities. For example the emulator task for a language runs at the lowest priority and the timer task that requests display memory refresh runs at one of the highest priorities. Control of the machine typically switches from task to task. The priorities are assigned based on the latency times that each can afford to undergo. Depending on the bandwidth requirements of the devices these tasks service, their priorities and degree of support by the processor microcode, they will employ one of three communication paths as shown in Fig. A6. The Alto suffers from small memory size and address space. These resulted primarily from the fact that the lifetime of the workstation was longer than planned.

LOGICAL INTERFACE LEVEL

PHYSICAL CONNECTIONS	BCPL or Asm (Loads & Stores)	Emulator or Timcd Task Microcode	Private Task Microcode
Parallel I/O Port	Impact printer Prom programmer CPU debugger XY input tablet Cassette tape	Switch welder Low speed raster scanner raster printer	
Memory Bus	Keyboard * Keyset *	Terminal concentrator Modern interface	Medium speed raster scanner Console computer
Processor Bus		Mouse * Hardware Multiplier	Display * Disk * Ethernet* Arpanet 9 Track Tape High speed raster printer Audio Modern interface

* *Included in standard Alto*

Figure A6. Schematic illustration of the input/output attachments on the Alto workstation.

System Characteristics

Processor Subsystem

- 16 bit microprogrammed minicomputer
- 1 k PROM and 3 k RAM control memory allowing for multiple instruction sets to be resident simultaneously

Memory Subsystem

- 128 Kbyte primary memory expandable to 512 kbyte
- 2.5 Mbyte removable catridge disk
- No virtual memory capability

Graphics Subsystem

- 808 X 606 bitmap
- Special instruction for the transfer of blocks of bits within display memory and between display memory and primary memory
- Special routines for "painting" in characteristics
- Mouse pointing device

I/O Subsystem

- Parallel I/O port
- High speed devices interfaced to the processor bus
- low speed devices interfaced to the memory bus

Software

- Support for several languages, BCPL, Mesa, Smalltalk and Lisp
- Electronic Mail
- Utilities for simple line drawings, editors and text formatting

A.6 Star

The Star [SEYBOL] workstation by Xerox Corporation grew out of their experience with the experimental Alto workstation. The large base of users provided a great deal of input and its success led to the design and implementation of the Star. This workstation is primarily meant for an office environment. As a result a great deal of effort went into designing the user interface. The Star is intended for use on the 10 Mbs ethernet.

All software is written in the Mesa programming language. The Star comes with a variety of software packages aimed at making the use of the workstations easy for the novice user. For this reason the user interface often employs pictorial representations of most objects. An example is presented in Fig. A7. Files or folders may be selected with the mouse pointing device and their processing initiated. For example a user may want to look for a file and examine its contents, select a specific document and edit it. The selection of a folder may reveal a list of files. A specific file may then be selected, and so on. Extensive use is made of the graphic portrayal of objects and menus. Another example is in the use of virtual keyboards. Bit

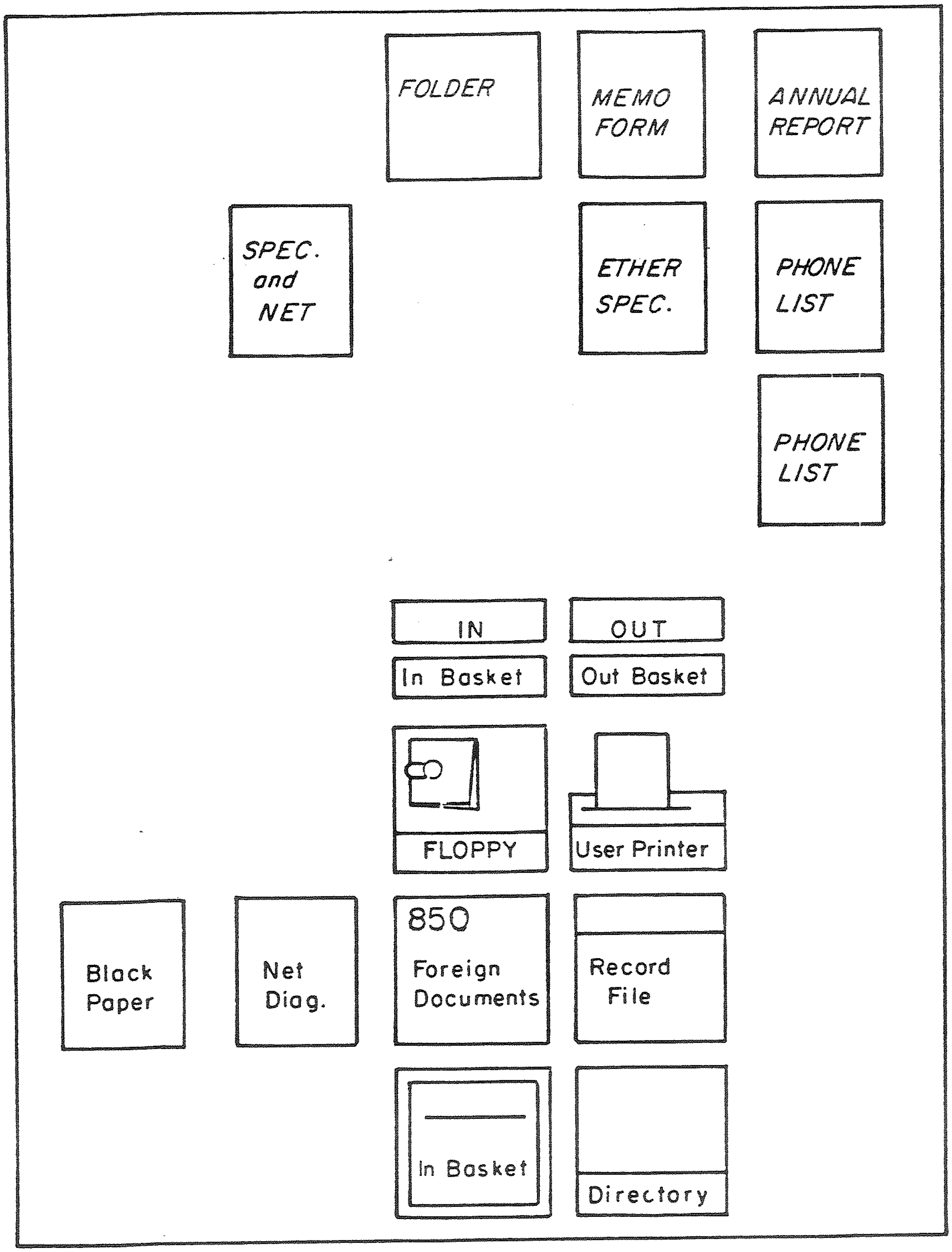

Figure A7. An example of the use of icons in the Star workstation.

fonts are stored for a number of special characters. When another character set is to be employed, the Star displays on the screen the current keyboard layout. Current software releases have included packages for processing of tables and records and include graphics packages for bar charts and pie diagrams. Grayscale capability is not currently supported. The use of this facility in addition to sophisticated editors and formatters makes for a very comprehensive office computer system. Files may be stored on the Stars local storage or in a central file server. The shared file server also serves as a center for receiving and sending electronic mail.

The main disadvantage of this system is its high price, in the neighborhood of $50,000 for initial configuration. This includes a $16,500 workstation, file server and high quality printer.

System Characteristics

Processor Subsystem

- Custom 16 bit microprocessor

Memory Subsystem

- 384 kbyte primary memory
- 10 Mbytes secondary memory standard
- Optional 24 Mbytes secondary memory available
- 1 Mbyte floppy disk available as an option

Graphics Subsystem

- 809 X 1024 bitmap
- Mouse pointing device
- Software defined variable size character fonts (upto 65000 characters can be stored at a time on the system corresponding to 256 character sets)

Software

- Provides pictorial representation of objects to be selected by the mouse pointing device
- Virtual keyboards
- Software for bar graphs
- Records and table processing
- Electronic mail

A.7 Dorado

The Dorado [CLARK,LAMP,LAMP80,MCDAN] is a compact high performance workstation designed to provide a hardware base for research to replace the Alto workstations at Xerox Corporation's Palo Alto Research Center. It is designed to be able to be attatched to the 3 Mbs ethernet. The emphasis in designing Dorado was on speed but it was not to be at the expense of flexibility. This resulted in heavily pipelined processor, memory, and dedicated intruction fetch unit to support fast execution. Additional features include characterization of four different instruction sets simultaneously (flexibility) and capability for expansion to accommodate forseeable advances in memory technology.

System Characteristics

Processor Subsystem

- 16 bit processor
- Pipelined instruction fetch unit which prefetches and decodes byte operands and whose operation is overlapped with processor operation
- 4 k cache to be expanded to 16 k
- hardware support for fast context switching
- Sufficient control memory to characterize four languages at once

Memory Subsystem

- 2 Mbyte primary memory to be expanded to 8 Mbyte
- 80 Mbyte secondary memory
- 22 bit virtual address space to be expanded to 28 bits

Graphics Subsystem

- High resolution bitmap driven display
- Mouse pointing device
- Grayscale and color display support

I/O Subsystem

- Full bandwidth communication between memory and processor possible through cache

- DMA access for high speed perpherals, e.g., color displays

A.8 SPICE

The development of a local network of workstations is being pursued at Carnegie Mellon University in Pittsburgh, PA. Their approach is aimed at a higher level, in that the design of the hardware organization is not their primary concern. The focus is on the environment that will be provided to the user of the system. The system is known as Scientific Integrated Personal Integrated Computing Environment (SPICE) [NEWELL,FAHL]. The design and contruction of personal computers is not a part of the project and SPICE is intended to be portable over workstations. Towards this end some effort has been focussed on determining the characterictics of the hardware required to support SPICE.

The processor should be capable of operation in the vicinity of 1 MIPS. It should be microcodable, providing at least 16K words of writable microstore. It should be possibe to have multiple resident intruction sets. This provides additional flexibility in addition to the benefits that automatically accrue from microprogrammability.

A high resolution bit map raster display with a resolution of at least 1024X1024 is required. Support for color is deemed a must with perhaps 4 bit pixel resolution providing 16 colors for output. In addition image generation aids are felt necessary for common operations and may be provided with hardware or microcode assist, e.g., simple transformations on portions of the display. To enhance the user interface a mouse pointing device should be provided for operations such as menu selection and selective editing.

SPICE is being developed with the goal of implementation by the mid 1980's. Keeping in mind the changes in programming environments that have taken place in the last few years, sufficiently large address spaces were deemed necessary to avoid outgrowing them to soon. To this end a minimum of 1 Mbyte of primary memory and 100 Mbytes of local secondary storage were deemed necesary. Virtual address spaces should be in the range 1-2 Bbytes. These numbers are for a typical SPICE machine and it is felt that capability for expansion should be provided for some machines.

Standard equipment for SPICE workstations should support audio and video I/O. This media intensive design coupled with requirements of transparent network access resulted in the choice of 10 Mbs bandwidth for the interconnecting network.

Each SPICE machine will host a kernel operating system which in combination with hardware and software will support SPICE. Features include very powerful interprocess communication mechanisms, transparent access to the network, multiprogramming and independent virtual memory for each process.

The software environment of SPICE is structured for maximum flexibilty and portability staying away from language biases, and the need for homogenity.

There are two levels of software below the user interface as shown in Fig. A8. One level consists of the set of facilities that are provided to support the worksatation hardware. Each facility accesses a portion of the next level which contains the environment of each facility. The user interface controls the environment of each facilty. The facilities themselves do not explicitly know how their input data is provided. This makes the software portable to many systems with quite different styles of interaction between the users. SPICE expects to have access to a large central file storage service on the network with local storage serving as a cache. The programming environment is expected

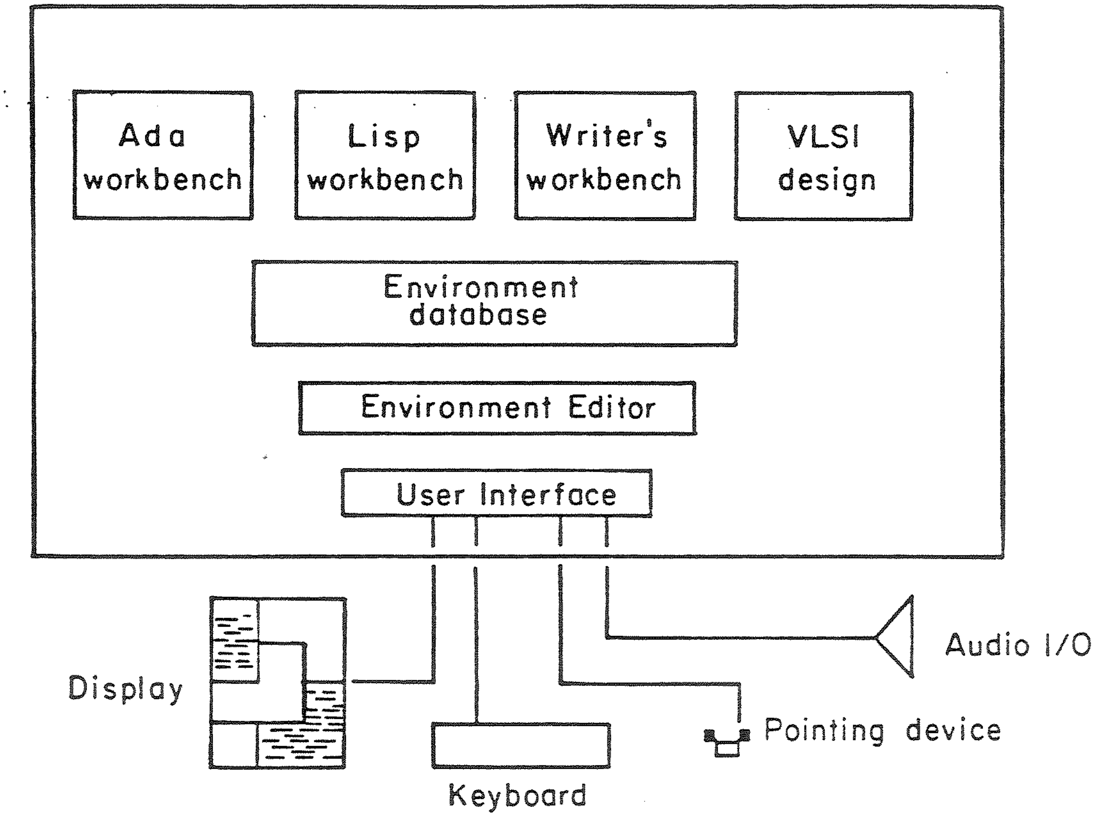

Figure A8. The levels of software organization in a SPICE workstation environment.

to include language support for Pascal, Lisp and Ada along with a screen oriented editor, text formatter and a facility for electronic mail.

A.9 Convergent Multifunction Workstation

This workstation [HUIE] is manufactured by Convergent Technologies, Santa Clara, CA. Up to 16 workstations can be clustered to share resources that is under the control of a master workstation. Each cluster workstation runs its own interactive program and communicates if necessary with the master workstation. Such a configuration is shown in Fig. A9. The operating system is not intended to be hardware independent - it is optimized for use with these workstations and can be configured and customized in the same modular fashion as the hardware. The operating system can be distributed because it is message based and not monitor based. The interprocess communication mechanism ensures proper ordering of the transmission and reception of messages by the various workstations. This communication system provides the workstation with its transparency. Access of files in the master workstation is identical with local access. This allows application programs to be written independent of clustered configurations and can be employed in stand alone mode or when interconnected in a cluster without any changes.

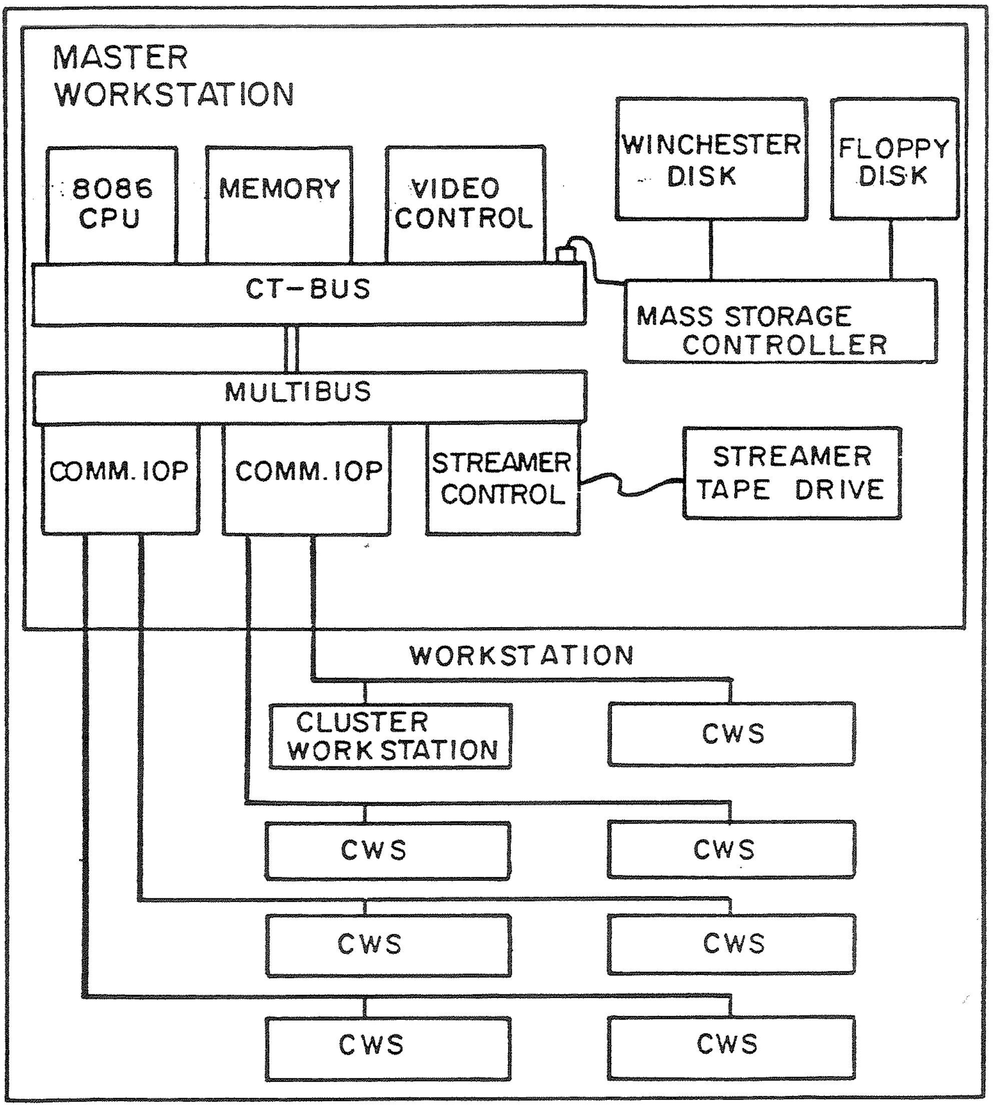

Figure A9. Convergent technologies multifunction workstation.

System Characteristics

Processor Subsystem

- 8086 16 bit microprocessor

Memory Subsystem

- 1 Mbyte primary memory
- Local secondary storage
- Optional floppy disk
- Virtual memory option that can be selected at the time program is linked providing an address space of 16 Mbytes

Graphics Subsystem

- 340 X 1024 bitmap
- Cursor
- Optional character set select
- Dedicated video controller
- DMA refresh

I/O Subsystem

- Proprietary CT bus for devices integral to the workstation
- Multibus for peripherals
- 8085 8 bit microprocessor as a communication

handler, each handler supporting communication with upto 8 cluster workstations

- Cluster workstations communicate over a 615 kbps network

Software

- Message based distributed operating system
- Configuration transparent to each workstation enabling application programs to run in stand alone configurations or network configurations wthout any change
- Operating system customized to the hardware