1323587

SCHULTZ, DARRELL E.

USE OF THE PERSONAL COMPUTER IN SOFTWARE ENGINEERING OF EMBEDDED SYSTEMS

COLEMAN COLLEGE M.S. 1984

University Microfilms International 300 N. Zeeb Road, Ann Arbor, MI 48106

USE OF THE PERSONAL COMPUTER

IN SOFTWARE ENGINEERING

OF EMBEDDED SYSTEMS

by

DARRELL E. SCHULTZ

A thesis submitted

in partial fulfillment of the

requirements for the degree of

Master of Science

Coleman College

March 1984

Members of the Supervising committee:

Preston Schiwitz, Chairman
Dr. Coleman Furr, CDP, Member
Joe Knox, Member

Darrell E. Schultz
3017 Mobley Street
San Diego, Calif.
Home 619-277-0111
Work 619-693-4147

TABLE OF CONTENTS

APPENDIX

LIST OF ILLUSTRATIONS

LIST OF TABLES

ABSTRACT

This thesis examines the advantages and disadvantages of a personal computer (PC) based distributed network and whether it can be applied to embedded systems development. The four primary aspects examined are the benefits to be derived from processing distribution, the hardware components necessary to support a PC-based local network, the software development tools available and the software development tools used.

Each of these aspects is examined from a PC-based viewpoint in an effort to determine whether the system can be used in a software engineering laboratory. To be usable, the PC-based system must demonstrate inter-connectivity capability with the large-scale Central Processing Unit (CPU) and must add capabilities to the existing system without eliminating any of the existing software development tools.

The thesis concludes that while hardware capabilities presently exist, the software development tools are not totally available.

I

INTRODUCTION

The industry-acknowledged advantages of distributed processing include overall performance improvement, lower cost of communications facilities, data sharing, and improved reliability and flexibility. The advent of the personal computer has opened the way for distributing the processing load down to the desktop level. One viewpoint is that, while the business community is accepting the new expanded role of personal computers with enthusiasm, the major developers of embedded software systems are shunning PC's in favor of remaining with the large-scale mainframe and non-distributed processing.

Assuming that this viewpoint is correct, this question arises: with the users in business embracing the new technologies, are the embedded systems developers, such as Hughes Aircraft, going to remain with what appears to be past technology? Additionally, in what direction are the other companies within this specialty headed? This situation is doubly curious since, in the development of embedded systems, the equipment is generally on a smaller scale than large business systems and operates with non-standard software and protocols. Of all the diverse areas within the data processing industry, development of embedded systems appears to be a natural area for personal computers tied into a distributed processing system.

The Research Question

Accordingly, the basic question this thesis will attempt to answer is:

Are the necessary hardware and software components available to enable the development of embedded systems with a distributed network of personal computers?

Subsidiary Questions

To answer this question, the following subsidiary questions must be answered:

A. Why replace the existing system?

B. What tools must the system support that are presently available to the programmer on the existing system?

C. What additional tools would the PC make available?

D. What advantages and additional assistance would the programmer receive from the personal computer?

E. What disadvantages and additional problems will the system create?

Hypotheses

A. It is technically feasible to replace the intelligent terminals with personal computers.

B. A PC-based distributed local network would improve performance over the existing system for developing embedded systems without degrading the existing software engineering system.

To prove or disprove the hypotheses, four aspects of embedded systems development are discussed in this paper. These are:

A. The relative advantages and disadvantages of distributed processing to provide a reason for the change. This area is discussed in Chapter II, titled Benefits of Distributed Processing.

B. The hardware capabilities of personal computers and local area network equipment. This will be used to determine whether the equipment is available which is capable of operating as a local area network in a VAX/VMS environment. This is discussed in Chapter III, titled PC-Based System Model Description.

C. The software development tools available at Hughes Aircraft and in the industry. This is discussed in Chapter IV, titled Software Development Tools Available.

D. The use the software tools are receiving. This area is discussed in Chapter V, titled Software Development Tool Usage.

Terms

Development tools are the set of procedures and software programs used to support software engineering. Tools include editors, word processors, compilers and graphics generators, among others.

Distributed Processing is processing performed in parallel by many processors. The processors are geographically dispersed but loosely coupled.

Embedded Systems are software programs which control the operation of a specialized hardware device not normally associated with data processing. Devices include radars, weapons systems controllers and radios, among others.

Hughes Aircraft Company, as used in this thesis, specifically refers to the Software Engineering Laboratory located in San Diego, California. The Laboratory is an operating unit of the Software Engineering Division, Ground Systems Group, Hughes Aircraft Company. The Laboratory is responsible for developing embedded software systems in support of the manufacturing divisions of Hughes Aircraft.

A Local Area Network (LAN) is a low-cost, high-speed transmission medium within a restricted area such as a single building or a small cluster of buildings.

Personal Computers (PC) are low-cost computers based on a microprocessor chip. They are somewhat portable, personally controllable and fairly easy to use.

Software Engineering is the set of disciplines used for specifying, designing and programming computer software to produce software that is reliable, economical, efficient and that meets the user's requirements. Methodologies associated with software engineering include structured programming, structured design and analysis, and the tools to support these.

Limitations

First, a delimiting factor is that this thesis is specifically targeted to the Software Engineering Laboratory in San Diego. With the scope limitation, it may or may not be applicable either to other software development facilities or to other operating units of the Hughes Aircraft Company.

Second, it is not cost effective to build the model envisioned by the thesis to prove or disprove the hypotheses.

Third, the number of expert opinions is limited to the San Diego area embedded systems developers.

Finally, to preclude having to evaluate one personal computer over another, only the personal computers marketed by the Digital Equipment Corporation (DEC) will be considered for incorporation into the model. DEC supplies the bulk of existing computer hardware to Hughes Aircraft.

II

BENEFITS OF DISTRIBUTED PROCESSING

This chapter discusses the pros and cons of distributing data processing requirements and sets forth some objectives for a distributed network. In addition, the advantages and disadvantages of distributed systems are discussed.

Existing Configuration

The existing software development system at Hughes Aircraft follows a centralized, multi-user pattern as presented in the much simplified schematic diagram in Figure 1. The VT100 terminals are tied into a multiplexor through the Teltone DCS-2B Data Carriers. The multiplexor ties into the Gandalf Data Switch at Fullerton for final routing to the programmer's account on a large-scale CPU, usually a DEC VAX 11/780. Dependent on the project the programmer is assigned to, the programmer's account can reside in Fullerton, Irvine, or San Diego, among others. Logging on the system presents the programmer with three potential bottlenecks. The first is getting on the multiplexor, the second is getting connected at the Gandalf switch, and the third is being accepted for log on by the VAX operating system. The programmer will not be accepted for log on if the maximum number of users is already on the system, if the system is being used for large scale testing, or if there is classified processing in progress.

The concept this thesis will attempt to develop is to replace this centralized system with a fully distributed software development system hosted on personal computers

(PC's). These PC's would be connected in a local area net (LAN) with access to the VAX via a high speed communications server in the LAN. The VAX would serve as configuration controller, maintain common programs, and keep the main storage files as well as execute large-scale programs which are beyond the capability of the PC. The hardware aspects of the proposed configuration are detailed in Chapter III.

The reasons for or against a PC-based distributed network parallel the reasons of distributed versus non-distributed databases. The only difference between the two is the degree of distribution. In a PC-based system, the distribution is much greater.

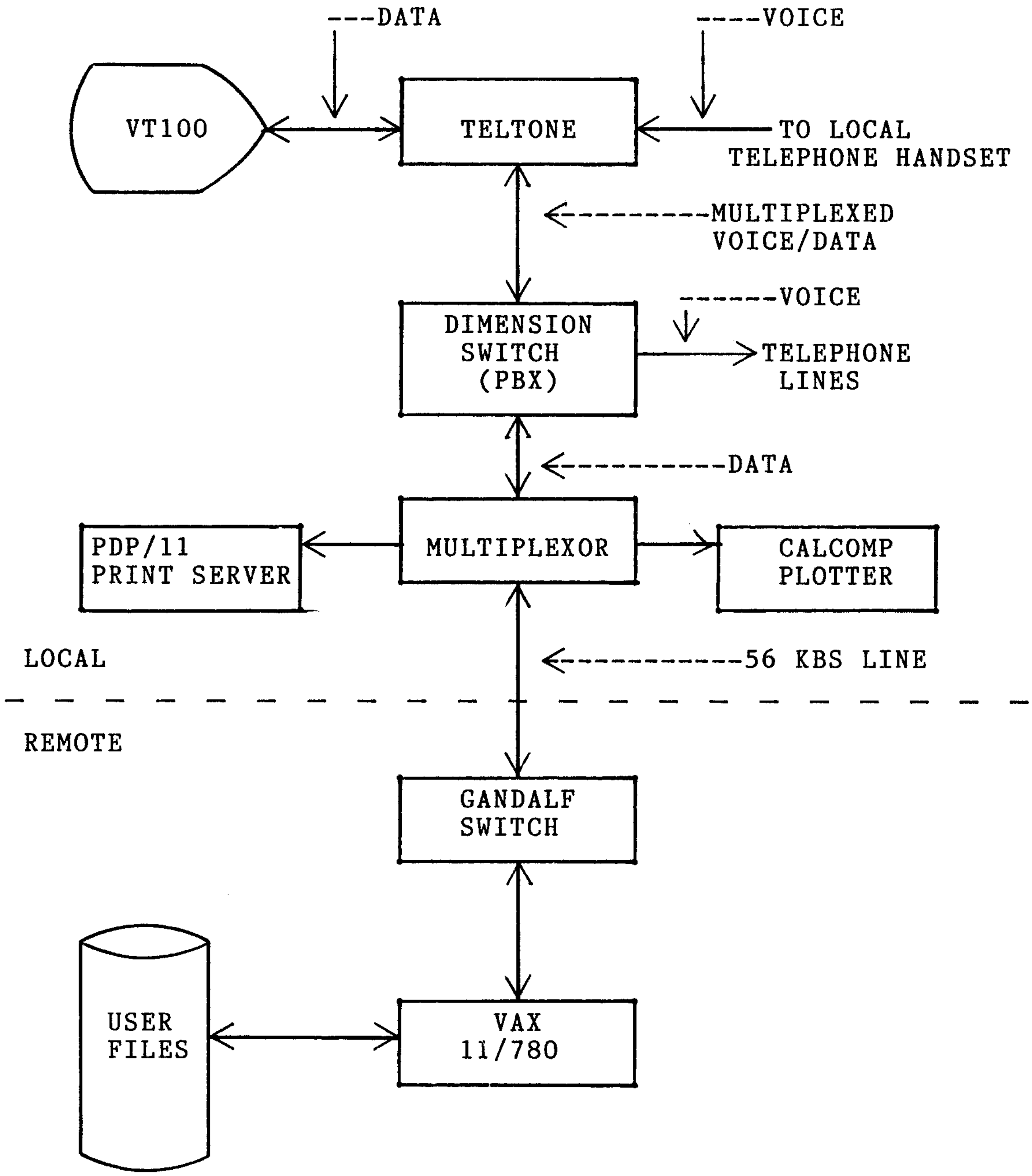

Figure 1

Software Development System

Hardware Configuration Simplified Block Diagram

The advantages of distributed processing have been well documented from its original conception through the existing systems today. Walsh (5:202) defines distributed processing as a processing environment in which two or more computer equipment configurations, remotely located from one another, are interconnected. In concept, both distributed processing and distributed databases provide the best of all worlds; that is, local processing where local processing is practical and efficient as well as centralized processing where that is practical. In the embedded systems development environment of Hughes Aircraft, this concept can be implemented readily since data normally belongs to the developing programmer until late in the development cycle. The database is segmented by programmer account, and each programmer maintains his own account. Whether the data resides in a central database or a local database is immaterial in embedded systems development.

Objectives

Sprunk (3:2) presented two basic types of objectives for distributed processing: technical and organizational. The technical objectives are primarily concerned with the distribution of hardware, and the organizational objectives are primarily concerned with the distribution of data processing functions. In a normal data processing environment, there is a distinct separation; however, in embedded systems development, the two are mostly synonymous. Accordingly, the objectives listed will apply equally to a technical as well as an organizational point of view. In other words, changes in the technical side will not require corresponding changes in the organizational makeup of Hughes Aircraft. Any organizational impact normally expected due to processing distribution should not occur in this instance.

As previously stated, the anticipated goals and objectives of distributed processing are basically the same as those of a distributed database. The main objectives of a distributed database are performance improvement, lower costs, data sharing, reliability and flexibility. These are also the objectives of distributed processing.

One objective for distributing databases is to improve the performance of a system in an effort to obtain faster response to programmer requirements. As a rule, processing requirements in a locally situated personal computer configuration is more rapid than accessing the mainframe at a remotely located computer configuration.

A second objective is to lower costs by reducing the number and the usage of communications facilities. Periodic transmissions of up-to-date information from a central site to local sites using a single multipoint facility, for example, is less expensive than having several points, each with a separate communications line and modems to a central site. Additionally, any transactions requiring the use of the communications line can be handled by the personal computer when the programmer has completed his immediate task.

A third objective is to permit the sharing of data among several geographically separated sites. In a distributed network a database may be fragmented functionally, and yet information can be made available to any computer configuration in the network either through data transfer via a local or leased telecommunications network or, in the case of the personal computer, by physically transferring diskettes.

A fourth objective is reliability. In most distributed arrangements, failure of a single-computer configuration does not stop the operation of the remaining configurations. With PC's, processing can continue to some degree as long as any one PC is still operating.

Another objective is flexibility. Related to reliability, flexibility allows other segments of a distributed configuration to assume some, or all, of the functions of the failed network component. Again, with PC's, as long as power is available, some processing can continue.

Advantages and Disadvantages

Why does distributed processing appear so attractive? Some advantages of distributed processing which are in line with the cited objectives include the following:

A. Response time (the length of time between entering a transaction and receiving a response) is potentially minimized and is not dependent on system loading.

B. Failure of system components have less impact on the entire system. In the existing system, failure of a key component can stop the productive work of an entire project or the entire professional staff.

C. Significant portions of the systems software overhead, which is necessary to support elaborate multiprogramming in a large-scale computer configuration, can be eliminated.

D. Communications costs may be reduced due to the emphasis placed on the local processing. Systems development work normally processed by a central computer configuration can be processed locally. Reducing the number of programmers contending for access can reduce the number of communication lines and modems, thereby reducing costs.

E. A degree of control over applications lies with each programmer. Scheduling, for example, is simplified. Programmers can schedule their own jobs locally, and they do not have to compete for resources with several other users of a large centralized computer facility.

Some of the disadvantages of distributed processing and databases are:

A. Overall management of a distributed system is relatively difficult. While it is true that local management is simplified, keeping on top of the entire network and all its facilities is a formidable management task.

B. The technology, while available in electronic components and software modules, has to be integrated for a distributed system to function properly. Although many of these components and modules are potentially compatible, a significant amount of time-consuming and painstakingly detailed technical tasks is necessary to make them work together.

C. In a sophisticated database/data communications environment, intersystem problems, such as deadlocks or deadly embraces, need to be addressed and resolved. This particular problem is minimal in embedded systems development since each programmer maintains his own unique data items.

D. Potential communications costs savings are offset by increased costs in duplicated equipment, programs and data.

Summary

This chapter discussed the pros and cons of distributed processing and set forth the objectives of performance improvement, lower costs, data sharing, reliability and flexibility. Whether or not a PC-based system can meet these objectives will be discussed in Chapter VI. From the cited advantages and disadvantages, however, distributed processing is preferable to non-distributed processing for software engineering.

III

PC-BASED SYSTEM MODEL DESCRIPTION

This chapter discusses the hardware-related aspects of the PC-based model envisioned to replace the VT100-based system presently in use at Hughes Aircraft. The three component parts of the model that will be described in this chapter are the personal computers (PC's), the local area network (LAN), and the protocols necessary to enable them to operate effectively with the VAX computer. To enable these component parts to function properly, compatibility of equipment is one of the major factors which dictates the choice of equipment. To re-emphasize a limiting factor stated in Chapter I, this model is based on DEC equipment since DEC provides the bulk of off-the-shelf equipment to Hughes Aircraft. For a model to have a chance to appear functionally compatible, the state-of-the-art in the market today mandates the use of a single manufacturer's equipment to the greatest extent possible.

Personal Computer

DEC is presently marketing three PC models, the Rainbow 100, the Professional 325/350 and the DECMATE II. According to Marbach (2:8), the Rainbow is the most traditional personal computer. Inside are two processors, an 8-bit Z80 and a 16-bit 8088. Because the state-of-the-art demanded a 16-bit processor and because there is a wealth of software written for an 8-bit processor, DEC chose to include both for its first entry into the personal computer market.

The Rainbow 100 is able to run the industry standard 8-

bit operating system, CP/M, written by Digital Research, Inc. It will also automatically run the newer CP/M-86 16-bit version of this operating system. Using a feature called SOFTSENSE, the Rainbow 100 will automatically run application software on the right processor without the user having to know which processor is being used. In fact, it is possible to mix and match CP/M and CP/M-86 programs and files on one diskette.

Because of the Rainbow's ability to run much standard micro software, it should be able to mimic any of the more popular machines and therefore have a ready-made software market. Unfortunately, the Rainbow diskettes are anything but standard, and transferring software to these factory-formatted disks will take some time and logistical planning.

MS-DOS, an operating system from Microsoft, is available for the Rainbow. This is a very popular operating system for the IBM personal computer, and its availability should make running many programs written for the IBM PC easy to do on the Rainbow, given the proper diskette.

Memory for the Rainbow comes standard at 64K with add-ons available from DEC and third party vendors for up to 256K maximum. Other options include extended bit-mapped graphics, dual floppy disk drives, Winchester disk drive, color monitor and any printer using the standard serial RS-232 protocol.

At first look, the Rainbow 100 is a full-featured top-of-the-line personal computer. From a hardware point of view, Marbach liked the Rainbow because of its display, its keyboard, and a system unit that sits vertically on the floor. The dual processors allow both 8-bit and 16-bit software to run, and when the software suppliers catch up, the Rainbow should be able to run most of the current offerings.

Much of the internal hardware and expansion capability is unknown. DEC is not doing a good job of telling everyone how to add peripherals to their computers. Dealers are applying "reverse engineering" to determine how to attach the peripherals.

Digital has always been sympathetic to people who needed to attach peripherals to their computers, but this may not be the case with the personal line. While most personal computers make it easy to attach peripherals, DEC may be making it too difficult. The add-on market has been a large driving force in the success of many personal computers.

For the Rainbow, the keyboard, display and system unit are on the plus side, with poor hardware add-on capability on the negative side.

The Professional 325 and Professional 350 make up the first two entries into DEC's top-of-the-line personal computers. The main processor in the Professional series is the tried and true PDP-11, perhaps the most widely made computer ever. The PDP-11 is a 16-bit processor which was introduced in 1970, more than thirteen years ago.

The Professional series computers use the P/OS (Professional Operating System). This was developed specifically for these personal computers and is a derivative of RSX-11. It requires no system generation and starts automatically when the system is booted. As options are added to the processor, there is a simple update procedure to let the operating system know what has been done.

P/OS uses menus to make the interface between it and the user simple and straightforward. There are two different modes to P/OS. One is the user environment, and the other is for programmers and software developers. Each system has features tailored to the particular application so that programmers get utilities and users get a menu structure.

A standard feature of all Professionals is a computer-based instruction program. This training aid makes use of the "help" key when needed and guides the user through a training procedure from the keyboard.

Physically the Professional series looks just like the Rainbow and DECMATE II. Inside there is the F-11, 16-bit PDP-11 computer, a full 22-bit addressing capability, a communications port, a printer port, a dual diskette drive and the P/OS. A 10-megabyte Winchester disk drive is available as an option.

Since all the DEC personals are dual processor computers, the Professonal series has a CP/M option. The 325 has only one slot inside for expansion, while the 350 has four available slots. The 325 can be upgraded in the field to a 350 with an expansion kit. While the memory is currently limited to 512K, the machine has the ability to address much more.

The extended bit-mapped graphics option is available and takes up one of the expansion slots. The optional color monitor will require the extended bit-map graphics option.

From the research conducted, the Professional-series PC has the greater capability for use in a software engineering facility. One major drawback that has to be considered is the matter of cost. Although cost has not been considered as a factor in the development of the PC model, in this particular case it is an overriding factor. The Rainbow 100 PC, at approximately $3,000, is 30 percent more expensive than the VT100, at approximately $2,000. The Professional-series, on the other hand, starts in the $10,000 range and is at least five times the cost of a VT100. Another factor that favors the Rainbow 100 is that the Professional-series appears to be based on outdated technology. This problem will be discussed more in Chapter VI. Accordingly, although the Professional-series appears to be the obvious choice, the PC-based model is based on the Rainbow 100.

The DECMATE II is billed by Digital as an office system computer. Although it is primarily a word processor, it also doubles as a real data processing machine. The original PDP-8 with the COS-310 operating system was a commercial DECSYSTEM not too long ago, and there are many of them still running.

The DECMATE II is a new set of clothes for a tried and true product. Digital's much-copied word processing software is good enough to be carried forward on the PDP-8 and included in the personal computer package. Communications to and from the DECMATE II are carried over from the earlier

DECMATE I. The DEC-developed DX (document transmission), AX (automatic transmission), and CX (character transmission) protocols offer full data exchange capability with other compatible computers.

DEC is also offering a variety of business software, including the Winchester disk-based Digital Accounting System and the diskette-based Business Accounting System. The DECMATE II makes a very nice office word processing machine and should compete well against other word processors.

In the real office automation world, the DECMATE II with its multiple personalities (WPS-8, the word processor; COS-310, the PDP-8 operating system; and eventually a CP/M capability) will fit in as some type of workstation with DEC's larger computer systems, particularly the VAX 11/700-series. At the present time, there are five stand-alone DECMATE II systems installed at Hughes Aircraft.

Local Area Network (LAN)

The options involved in local area networks are not as straightforward as selection of a PC. This section will briefly discuss some of the aspects of the LAN and how it relates to the model envisioned and how a typical microcomputer based LAN would be established and used.

According to Computerworld magazine (6:99) the original LAN's were minicomputers communicating with each other in a distributed network or under control of a mainframe, so there was a somewhat limited application. The increasing acceptance of microcomputers in a variety of environments has renewed the interest in LAN's and turned LAN's into one of the buzzwords of the present era.

Prior to the discussion of LAN's, some definitions are presented here as they apply to this section:

A. LAN: The classic definition of a Local Area Network (LAN) is a data communications network, typically a packet communication network, limited in geographic scope, which provides high-bandwidth communication over inexpensive transmission media.

B. PROTOCOL: Defined instructions to enable devices to communicate; i.e., the way of doing things.

C. TOPOLOGY: The actual physical relationship of the "nodes" in the network. The relationship may be:

. UNSTRUCTURED GRAPH topology which is usually free-form built; i.e., added on bit by bit. The routing requirements are somewhat complex.

. STAR topology which has a central processor making decisions and doing the routing.

. A RING/BUS Network which eliminates the central node and simplifies routing. The node must simply receive from one node and send to next node; however, a CONTROL structure is needed for the RING/BUS. There are two types of controls. One type is Control token, where the node with the token can transmit. The other type of control is the Message Slot, where the node marks an unused slot for node transmission, is assigned message slots for transmission dependent on traffic load, or where the node transmits when needed but can recognize and recover from collisions.

Some of the transmission media available for establishment of the LAN are twisted pair, coaxial cable or fiber-optic cables and combinations of one or more of these. They can either be dedicated to only the LAN, or they can be piggy-backed onto telephone lines such as the TELTONE system presently in use at Hughes Aircraft.

Microcomputer Networks/Local Net Protocols

The first of three common topologies is simple point-to-point communications. Device A transmits an asynchronous, ASCII/TTY data stream through a modem across the line to a second modem connected to Device B. The Device A modem translates the data stream into an analog telephone signal. The modem at Device B demodulates that signal, turning the wave form back into digital information.

Typically, however, a basic microcomputer system does not include this kind of communications capability. Instead, it must be upgraded with an expansion board in order to communicate with the outside world. This type of arrangement normally permits either 300 or 1200 BPS service. In addition to add-on hardware, communications are controlled by utility software routines. At the simplest level, a communications utility program enables the user to select a file from a diskette and to format it for transmission to a remote location.

The most common protocol is TTY, long established as an asynchronous protocol standard. Independent software houses offer a wide variety of other packages that include more sophisticated error handling, data capture and file translation features.

In a second common situation, the microcomputer seeks information from a minicomputer or mainframe. This access is usually gained through a communication controller attached to the host processor to coordinate terminal activity. While it is possible to transfer files between the mainframe and micro, in most cases the formats of the large system files are incompatible with the microcomputer files and therefore are unusable.

Because of this incompatibility, current micro-to-mainframe communication usually involves the microcomputer simply emulating the capability of a large-screen terminal such as a DEC VT100.

There are many hardware and software products designed to turn the micro into a terminal. The final result of this transformation is that the micro uses application programs such as payroll data entry and interactive data base query running on the large system. Chapter VI discusses the specific applications for this model.

A third possible network configuration is designed to provide shared resources to a local work group. Historically, the first networks of PC's grew from the need to share access to expensive Winchester disk drives. It was difficult to cost justify a storage device that often sold for more than double the price of the micro itself.

One of the first micro networking systems was called Constellation. This product used multiplexing techniques and a star topology to share a multi-megabyte Winchester. The multiplexor polled each micro in a time sequence to assess demand for disk access. Although this technical strategy of multiplexing achieved the objective of sharing Winchester resources, it soon became clear that more power and flexibility were needed in certain office environments.

As an enhancement to this configuration, the next system introduced was a system called Sharenet which used a Network Manager designed around the Motorola 68000 microprocessor and a star network topology. Each micro was hooked to the network by a network interface device, with each network device being capable of supporting up to 24 micros.

Today, Sharenet offers four major capabilities. The first is to provide access to a shared hard disk. Sharenet offers both mass storage and concurrent file sharing with information security options. This capability frees the user from the limitations of floppy disks and maximizes the availability of information for multiple network users.

With Sharenet, any two micros can have point-to-point communications through a program built into the Network Manager. It is possible to transfer a file or a message directly from one user to another without transiting through a storage medium. Sharenet also offers an electronic mail capability.

The fourth major characteristic of Sharenet is shared access to printers through print spooling. Sharenet can support up to five printers simultaneously. The print spooling software assigns document numbers, determines print sequence and manages other job control requirements.

The next evolution in LAN's is the Ethernet, which is a microcomputer implementation based on Xerox Corporation's Ethernet Standards.

Unlike the Sharenet Network Manager/Star topology, Ethernet uses a data bus using the Carrier Sense Multiple Access/Collision Detection (CSMA/CD) standard. CSMA/CD means that when a station wants to transmit data, it listens for network activity of messages generated by other devices before beginning a transmission. A collision is detected when there is no message of acknowledgement from a receiving station or when the information is garbled. In this case, the stations involved wait a random time interval before broadcasting their data again.

Use of the CSMA/CD protocol provides many advantages. The primary advantage is that a network station transmits only when needed, leaving the net free of a great deal of control overhead. Secondly, the PC can be taken off-line for other uses and does not have to be set up for communications at all times. Since the Ethernet standard represents the latest state-of-the-art system available for a LAN, and since there is no existing equipment available at Hughes Aircraft to support a LAN, an Ethernet LAN was selected for the PC-based model.

One product line for connecting non-networked devices is the Bridge Communications Ethernet System Product line. Their product line consists of three communications computer-based products which permit integration of existing equipment into an Ethernet network while providing vendor independence. The PC-based model will incorporate one of the three products, the Communications Server. The Communications Server can link non-networked devices (terminals, modems, printers, word processors, and PC's) with standard RS232/423 serial interfaces to the Ethernet bus.

The Communications Server (CS) is a modular communications processor designed to provide a method of interconnecting non-networked devices over an Ethernet local area network. The CS can support up to 32 devices which have an RS-232 serial port. The CS may also be configured to support synchronous devices, IEEE 488 compatible systems or multiple high-performance peripherals with Versatec "green sheet" parallel interfaces. The Communications Server is compatible with the Ethernet Version 1.0 and IEEE 802.3 standards at the physical and data link layers. The higher level protocols (ISO 3 - 5) are fully compatible with the Xerox Network System product line. A custom network can be configured using multiple CS's to interconnect equipment to create a distributed data switch in a multivendor environment.

Some of the additional features of the Communications Server are:

. An RS-232 port which can support data rates from 50 to 38.4K bytes per second in an asynchronous mode and up to 300K bytes per second in a synchronous mode.

. Fully configurable port, system and network parameters. Port parameters include individually selectable port data rate, parity, flow control, start/stop, data bits, and autobaud options. System parameters include buffer size, command levels, and rotaries. Network parameters include access control and distributed naming.

. Configurable as a dedicated host interface or peripheral device server with high speed parallel I/O interfaces that support IEEE 488 compatible host or instrumentation systems. The server supports Versatec "Green Sheet" compatible electrostatic or laser printer and plotter devices with data transfers up to 100K bytes per second.

. Multiple 68000 (16/32 bit microprocessor) architecture provides a high performance Ethernet controller, protocol processing and device I/O (200 packets per second aggregate).

. Xerox Network Systems high level protocols (ISO 3 - 5) provide end-to-end reliability, routing, sequencing, flow control and session multiplexing.

. Network management facilities for performance monitoring, runtime error logging and hierarchical configuration control. The CS supports fully distributed logical naming service.

PC-Based System Model Summary

This chapter discussed the hardware aspects of the PC-based model which includes the PC and the communications devices necessary to support a local area network. Figure 2 summarizes the items covered in this chapter. The

Communications Servers are tied together over an Ethernet bus with the VAX, Rainbows and DECMATEs being serviced by the Communications Servers. It can be concluded that from a strictly hardware point of view, the technology exists to replace the VT100 based system at Hughes Aircraft with a distributed system based on personal computers.

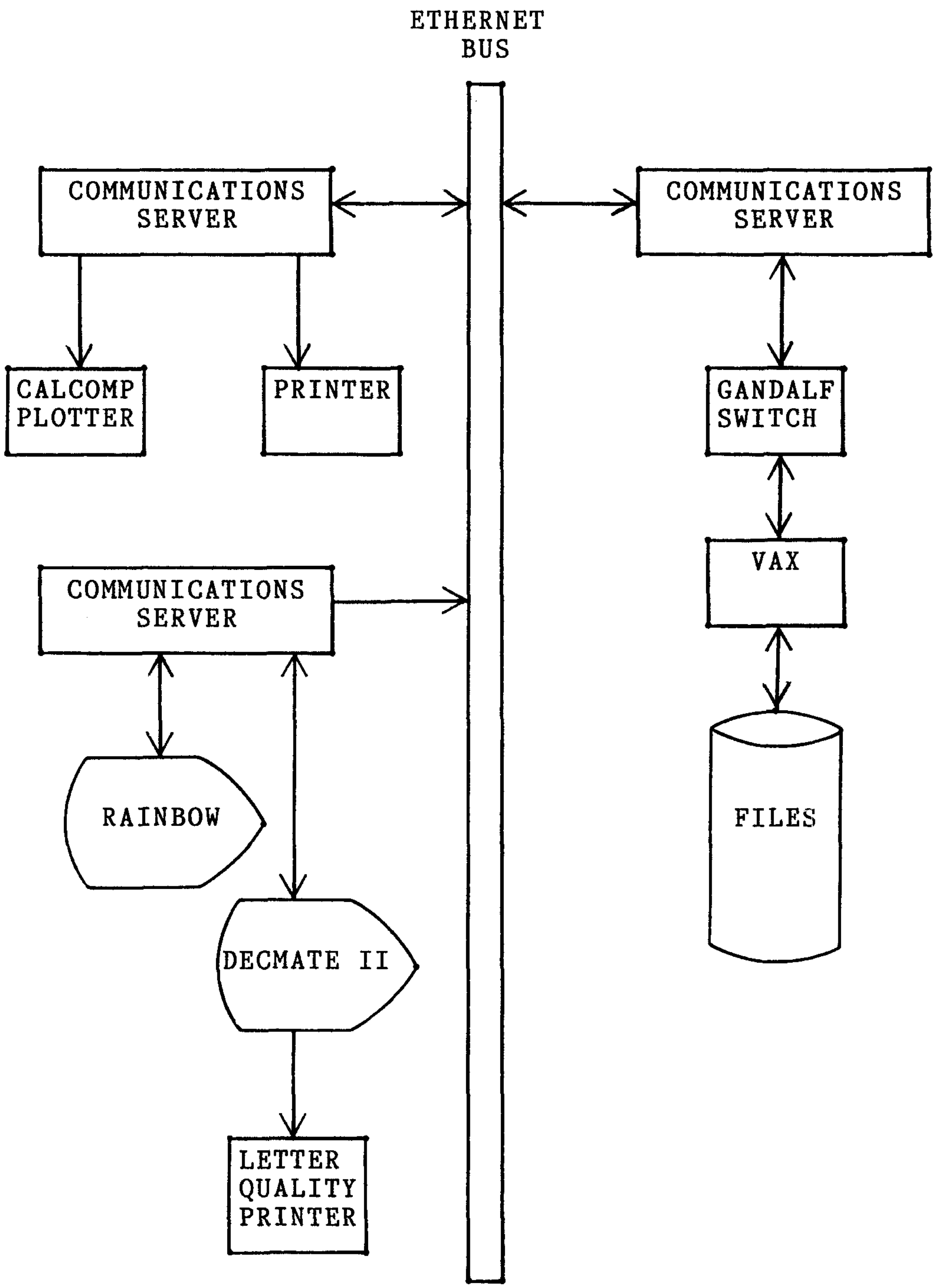

FIGURE 2

PC-Based Distributed Network

Simplified Block Diagram

IV

SOFTWARE DEVELOPMENT TOOLS AVAILABLE

This chapter discusses the tools available for embedded systems development from the aspects of what is available in the industry and what is specifically available to the programmer at Hughes Aircraft.

In a effort to answer the questions of what software development tools are available, what the capabilities of these tools are, and where the tools can be obtained, Houghton (1:63) established a database of software development tools at the National Bureau of Standards' Institute for Computer Science and Technology (ICOT). According to Houghton, software development tools have grown in complexity in recent years, as have other software applications. Most early tools performed a single function and were very simple to describe. Tutorial papers of earlier years usually classified tools according to their function. These simple tools have since given way to more complex tools with increased functionality, and major shifts are occurring in the types of tools being developed and marketed. For example, earlier development work was in compilers, debuggers, dump analyzers, flowcharters, and editors. Now development attention has shifted to software development systems, application generators, software engineering workstations, program generators, and programming environments. The capabilities of these latest tools make them much more sophisticated than their earlier counterparts.

The tools categorized by ICOT form the baseline from which to analyze software development tools available on PC's when compared to the tools presently available at Hughes Aircraft. In addition, the information compiled was used to develop the Industry Survey contained in the appendix of this thesis.

Categories

The tools available for software development fall into one or more of the following general categories:

A. Compilers

B. Source Program Analysis and Testing

C. Software Management Control and Maintenance

D. Requirements/Design Specification and Analysis

E. Program Construction and Generation

F. Software Modeling and Simulation

G. Software Support System/Programming Environment

The database established by ICOT discounted the normal program development tools, such as compilers and debuggers, but in order to properly evaluate the extent to which a PC can support software engineering, they are included as evaluation criteria.

The compilers available for software development are listed in the Industry Survey contained in the appendix. This study will place particular emphasis on ADA, ALGOL, C, CMS-2, FORTRAN, JOVIAL, PASCAL, and RATFOR as the programming languages normally associated with the development of embedded systems and as the languages presently available to Hughes Aircraft programmers. In addition to tools normally associated with programming, tools have been developed with the following features:

. Transformation features such as design language, usually psuedo-code, full screen or line editors, and optimization compilers.

. Static analysis features to provide data flow analysis and to measure the complexity of the program.

. Dynamic analysis features to provide path flow tracing, breakpoint control, logic flow tracing, data flow tracing, and symbolic execution. Dynamic analysis tools include simulators and model-building tools.

. Management features for configuration control, project management, database management, change control, test data management, files management, library management, and documentation management, as well as cost estimation and scheduling.

. Graphic outputs, either on a hard copy device, such as a plotter, or on the terminal itself. These include flow charts, HIPO charts, bar charts, milestone charts, and structure charts.

. Document and report generation facilities.

Hughes Programming Tools

For embedded systems development, the programmers at Hughes Aircraft have the full gamut of software development tools available (7:1). Table 1 contains a synopsis of the tools available categorized in accordance with the list in the previous section of this chapter. Following the table is an amplification of the capabilities and features of each tool listed.

Table 1

Hughes Aircraft System Development Tools

Function	Category
Full screen editor	E
Line editor	E
Compilers: ADA C CMS-2/MTASS JOVIAL FORTRAN INFORT RATFOR MACRO-11 PASCAL	A
Simulators	F
Configuration Control System	C
Text formatter	G
On-line structure charts	G
Management Information Report Generation	C
Software Design and Documentation Language	B
Utilities	A

The full screen editor allows editing on selected terminals. The user has a view of a full screen (approximately 22 lines) at any time during the editing process. Changes can be made directly on the screen simply by typing over the characters. Special function keys perform such operations as deleting characters, adding or deleting lines of text, moving lines of text and pattern searching. Commands are entered either with a command language, through function keys, or by combinations of both. Full screen editors are usually limited to a specific terminal type for complete operation. Hughes programmers have two screen editors available, EDIT/EDT and EDIT/EVT.

Using a line editor, the user operates on one line of the file at a time. A command language is used, allowing for listing, inserting, deleting and modifying text. Line editors normally will run on any type terminal. Line editors include EDIT/SOS, EDIT/FDL, EDIT/SLP and EDIT/SUM.

The compilers available to Hughes Aircraft programmers are:

. ADA, which is a highly structured language with strong data typing in the family of ALGOL and PASCAL. In addition to normal compilation functions, ADA is being developed to encompass the entire programming environment including loaders, libraries and editors, among others. The language is presently being developed as a Department of Defense standard for embedded systems. At the present time, only a training interpreter for the ADA language is available to the programmers at Hughes Aircraft.

. C Programming language, which is a general purpose programming language closely associated with the UNIX operating system. C provides only single-thread control structures such as tests, loops, groupings, and subprograms, but does not support multiprogramming, parallel operations, synchronization, or coroutines. C is a structured language similar to PASCAL, ALGOL, and PL/1, and has such language constructs as WHILE-loops, IF statements, FOR-loops and switch statements. Some of the data structures included in C are arrays, user-definable structures, and linked lists with pointers. C also provides pointers and the ability to do address arithmetic. It is a loosely typed language.

. CMS-2, also known as MTASS in the VAX environment, is a high level programming language combining some of the best features of CS-1, FORTRAN, JOVIAL and PL/1, while adding more extensive program error locating, correcting and testing features. It is designed primarily for real-time, command and control applications. It can also be used for data processing and scientific processing. CMS-2 is the present Navy standard language for embedded systems.

. JOVIAL is a high-level language geared for the development of large-scale command and control software systems. JOVIAL is a general purpose, generally machine-independent, procedure-oriented language. It describes data processing in terms of the logical operations required. It is a readable language using notations from algebra and logic expressed in self-explanatory English language terms. Some of the language constructs included in JOVIAL are IF-THEN constructs, FOR-loops, and procedures. JOVIAL also has the ability to handle table (array) data structures. JOVIAL is the present Air Force standard language for embedded systems.

. FORTRAN is a widely known general-purpose language best used for mathematical related problems. It is machine-independent and does provide some structured constructs such as DO-UNTIL, IF-THEN-ELSE, and iterative statements, but it does not provide case statements. In addition to integer data it also supports single-precision, double-precision, and character type data with the ability to handle a specific data type in array data structures.

. INFORT is an implementation of the general purpose language FORTRAN on the PDP-11. INFORT is similar to ANSI standard FORTRAN-66 with some additional features. Since INFORT uses storage space for variables only at the routine's runtime instead of at compile time like FORTRAN, it can save a large portion of storage space during execution. INFORT provides constructs such as DO-loops and IF-THEN-ELSE statements. It accepts integer, logical, single precision, double precision, and character data types in array data structures.

. RATFOR (RATional FORTRAN) is a FORTRAN preprocessor that allows code to be written in a structured language similar to PASCAL, ALGOL, and PL/1, having such language constructs as WHILE-loops, IF-statements, and FOR-loops, and having the source code translated into portable FORTRAN-66. RATFOR retains the merits of FORTRAN such as universality, portability and efficiency while providing a language that is easier to read and write.

. MACRO-11 is an assembly language used on the PDP-11 and VAX-11 computers. Since it is a low-level language, it is able to control I/O processing more powerfully than high-level languages (FORTRAN, C, JOVIAL, etc). Its code is also more efficient than high-level languages. MACRO-11 can simulate practically any data structure used in a high-level language. It provides a powerful macro defintion capability. One drawback is that one statement in a high-level language may take from one to ten or more statements in MACRO-11. Another drawback is that programs written in MACRO-11 will generally take longer to write than when written in a high-level language. MACRO-11 programs, like most assembly language programs, are not portable.

Simulators provide for the execution of target programs by emulating different machines. Loaders are provided for ULTRA-16 and JOVIAL object formats. Features include single step, breakpoints, register and memory display and modification, snapshots, dumps and memory search. Simulators can be used in either the batch or interactive modes.

The configuration control system (CCARS) is a set of processes designed to help projects control changes to source code. It provides facilties for storing, updating, and retrieving, by version number, all versions of source code modules, and for recording who made each software change, when it was made, and why. Program trouble report data is maintained interactively by the system.

The text formatter is a text formatting language designed to make documentation more efficient. It offers a variety of documentation styles including footnotes, multiple column output, automatic section sequencing, page headers and footers, right-justification, and table of contents. RUNOFF and TECO are available to Hughes Aircraft programmers.

The on-line structure chart program (AIDES) is intended to be used mainly as a documentation aid while the design function is still in the hands of the developing programmer, although it can be used for incorporation of structure charts in final delivery documents. The structure chart generator consists of two parts: a design function and a hard-copy function. As the names imply, the design function is used for creating or changing a structure chart interactively on a terminal, and the hardcopy function is used for obtaining a plot of the structure chart on a hardcopy device. Once a structure is stored in the data base, the programmer may recall the structure for modification, saving structure redraw effort.

AIDES will produce structure charts in response to user statements, handle design files, provide configuration control, and control the printing, plotting, and displaying of design information. AIDES is composed of eight separate tools. The communication of design information between the tools is accomplished by using a data base management system. Each tool views the information in a form dictated by a common data base schema for design files that is used by each tool. The AIDES tools are:

Create/Edit/Display

Predefined Report

Module Summary Report

Table Summary Report

Structure Chart Report

Copy

Delete

Update

AIDES is a Hughes Aircraft Company proprietary system which cannot be downloaded onto another system.

The Management Information Report Generation (MIRG) program produces the following reports:

. A milestone report which displays the status of entities (such as program modules) with regard to design, code, parameter and assembly test (P&A), and integration phases of software development. It reflects estimated and actual week numbers per entity for start and completion on all four phases of software development.

. A barchart schedule which displays the design, code, P&A, and integration status of each entity in a barchart representation. Up to 52 weeks of status information can be shown on a report.

. A statistical summary report which measures entity and program task status, as well as work in progress. It computes entity counts and percent completion status per development phase, based on project-specified classes of entities.

. A rate chart report which displays the rate of projected versus actual development for each phase of design, code, P&A, and integration. A single phase can be represented, or a composite of all phases may be shown.

All the reports within the MIRG program, except the ratechart, can be printed or listed on a hard copy terminal.

They may also be viewed through CRT terminals. The ratechart program has been developed to run on the Calcomp plotter.

The objective of the Software Design and Documentation Language (SDDL) is to provide an effective communications medium to support the design and documentation of complex software applications. This objective is met by providing the following: a design and documentation language with forms and syntax that are simple, unrestrictive, and communicative; a processor which can enhance the information content of the design data base and produce an intelligible, informative machine-reproducible document; and methodology for effective use of the language and the processor. The SDDL processor is written in the PASCAL programming language for maximum portability. It has been rehosted on several systems including the Univac 1100, VAX, Harris and AODC.

The purpose of SDDL is to provide a bridge between the software design developer and the design reader which will reduce effort and enhance effectiveness for both tasks. To achieve this goal, the processor must perform as many automatic functions as possible. It must support transition of the designer's creative thinking into an effective communications document. To the extent that this goal is realized, both the designer's and the reader's energies become freed to apply to their direct tasks.

The objective of SDDL is to support these communications requirements of the software design and documentation process. This objective is met by providing:

. A design and documentation language with forms and syntax that are simple, unrestrictive, and communicative.

. A processor which can translate design specifications, couched in SDDL syntax, into an intelligible, informative, machine-reproducible Software Design Document (SDD).

. Methodology for effective use of the language and the processor.

SDDL has been effective in meeting its objectives on tasks ranging from small, one-person informal projects to large projects of hundreds of pages of design which were published formally.

The SDDL program can be described simplistically as a language processor with built-in knowledge of, and methods for handling, the concepts of structure and abstraction which are fundamental to software development and documentation. The program is a batch processor which reads the design data base, reformats it to emphasize and clarify structure, compiles information about the data, and prints the resulting information-enhanced Software Design Document. The reformatting step uses indentation to show structures and flow lines to highlight module escape (left arrow) and module invocation statements (right arrow). The result of the reformatting step is topologically equivalent to a flow chart and has the added advantages of having regular structure, no constraints on the amount of information in a structure node or body, and prohibited use of spaghetti-like "structures." All of the SDDL output is produced in standard line printer format which can be output on a terminal or diverted for batch output on a high speed line printer.

VAX System utilities include the following:

. CMS, a code management system to assist in managing source and other text files of an on-going project.

. LIBRARY, which can replace a module in an object, macro, or text library, create or modify program libraries, and insert, delete, extract or list the modules or symbols within a library.

. LINK, which links one or more object modules into a program image and defines the execution characteristics of the image.

. RMS is the VAX-11 Record Management System which consists of generalized routines to assist programmers in processing and managing files and records. RMS provides transparent network processing between DECnet and VAX nodes. RMS also includes a set of macro instructions that can be used to intialize file control blocks.

. RTL, the VAX-11 Run Time Library, is a collection of procedures available to native mode images at run times. Description of the actual procedures available is beyond the scope of this paper.

Summary

This chapter discussed the tools available for embedded systems development. The specific tool categories available included compilers, source programming analysis and testing, software management control and maintenance, program construction and generation, software modeling and simulation, and software support systems/programming environment. The programmers at Hughes Aircraft have all tool categories available to develop embedded systems. Use of these tools is discussed in Chapter V.

V

SOFTWARE DEVELOPMENT TOOL USAGE

To determine factors which can be used to evaluate the feasibility of the PC replacing the smart terminal, the surveys in the appendix of this thesis were developed. The Employee Survey was distributed to the professional staff of Hughes Aircraft, and the Industry Survey was distributed to San Diego area embedded systems developers. This chapter outlines the results of the surveys.

Employee Survey Results

The Hughes Employee Survey was submitted to the professional staff members of the San Diego Software Engineering Laboratory. The survey excluded project and laboratory management personnel, administrative positions and on-site contractor personnel. There were 78 surveys distributed with 35 responses received. The responses were used to build the evaluation matrix contained in Table 2. The usage factors of Often, Occasionally, Rarely and Never were given weighting factors of 3, 2, 1, and 0 respectively.

Table 2

Tool Evaluation Matrix

	USED OFTEN TOTAL	USED OCCAS. TOTAL	USED RARELY TOTAL	USED NEVER TOTAL	WEIGHT
ADA	1	2	7	25	14
BASIC	0	0	1	34	1
C	0	3	4	28	10
MTASS/CMS2	11	3	1	20	40
FORTRAN	12	6	1	16	49
INFORT	0	0	1	34	1
JOVIAL	7	2	1	25	26
MACRO	0	0	1	34	1
PASCAL	1	3	4	27	13
RATFOR	0	1	1	33	3
CCARS	1	1	2	31	7
DEBUG	9	5	1	20	38
DUMP	1	2	4	28	11
PATCH	0	1	2	32	4
SIMULATOR	4	1	2	28	16
RUNOFF	0	4	8	23	16
TECO	0	0	1	34	1
EDIT/EDT	17	7	5	6	70
EDIT/EVT	15	6	4	10	61
EDIT/SOS	1	4	0	30	11
EDIT/FDL	0	0	1	34	1
EDIT/SLP	0	0	1	34	1
EDIT/SUM	0	0	1	34	1
MILESTONE	0	1	0	34	2
BAR CHART	0	1	0	34	2
STATISTICAL SUMMARY	0	1	0	34	2
AIDES	10	11	8	6	60
CMS	2	2	0	31	10
LIBRARY	2	6	2	25	20
LINK	8	4	1	22	33
RMS	2	0	1	32	7
RTL	1	1	1	32	6
SDDL	7	12	2	14	47

As can be expected in a Software Engineering Laboratory, the normal software development tools showed the highest usage factors. These tools included:

. The full screen editors, EDIT/EDT and EDIT/EVT.

. The major development languages, FORTRAN, JOVIAL, and CMS-2/MTASS.

. The documentation and design tools, AIDES and SDDL.

. The development assistance tools DEBUG and LINK.

One unexpected result was that all tools listed on the survey showed at least some rare usage. Table 3 shows the tools ranked by their weighted usage factor.

Table 3

Tool Use Ranking

Overall Ranking	Tool Name	Weight
1	EDIT/EDT	70
2	EDIT/EVT	61
3	AIDES	60
4	FORTRAN	49
5	SDDL	47
6	MTASS/CMS2	40
7	DEBUG	38
8	LINK	33
9	JOVIAL	26
10	LIBRARY	20
11	SIMULATOR	16
12	RUNOFF	16
13	ADA	14
14	PASCAL	13
15	DUMP	11
16	EDIT/SOS	11
17	C	10
18	CMS	10
19	CCARS	7
20	RMS	7
21	RTL	6
22	PATCH	4
23	RATFOR	3
24	MILESTONE	2
25	BAR CHART	2
26	STATISTICAL SUMMARY	2
27	BASIC	1
28	INFORT	1
29	MACRO	1
30	TECO	1
31	EDIT/FDL	1
32	EDIT/SLP	1
33	EDIT/SUM	1

Evaluation Criteria

Figure 3 is a graphic representation of the survey results by weighted usage factor. Viewed from this perspective, it is evident that for the PC-based system to be considered a viable alternative, it must support the following high usage tools:

. Full screen editors equivalent to EDIT/EDT and EDIT/EVT.

. Structure chart development tool equivalent to AIDES.

. Software design language equivalent to SDDL.

. FORTRAN, JOVIAL and CMS-2/MTASS compilers.

. Debugging and linking tools equivalent to DEBUG and LINK.

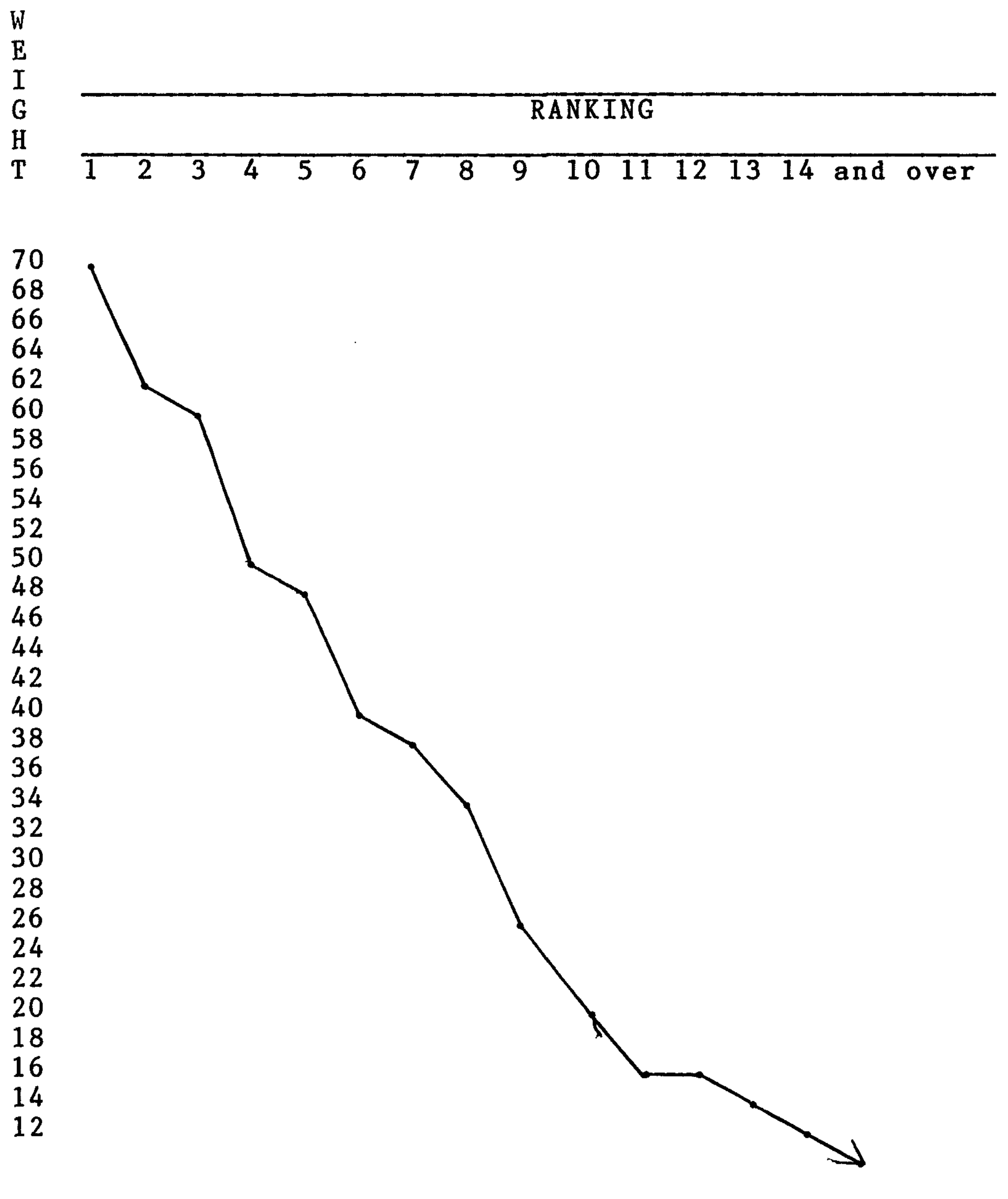

Figure 3

Usage Curve

Industry Survey Results

There were 82 surveys sent to 27 companies in the San Diego area. To the greatest extent possible, the surveys were addressed to project managers at the various companies, although this was not possible in all cases. The companies ranged in size from small one-contract firms such as Command, Control and Communications, Inc. to large manufacturing concerns, such as General Dynamics.

The purpose of the survey was twofold: to validate the survey conducted at the San Diego Software Engineering Laboratory and to support or refute the conclusions drawn from this paper.

Of the 82 surveys sent, only four responses were received. With this small response, no meaningful empirical data can be drawn from the survey. Since the survey was targeted at project managers, however, an assumption can be made that these are expert opinions and these opinions carry sufficient weight to be mentioned.

All of the companies that responded use DEC mainframes, with one specifically identifying a VAX 11/780. Three of the companies are presently using PC's in system development, but none of the PC's are in a local network configuration.

The stated advantages of the PC's include:

. No single point of failure in the development cycle.

. Better programmer attention due to improved response time.

. Portability and lower cost.

The stated disadvantages include:

. Logistics of coping with multiple hardware configurations.

. Misuse by employees is harder to detect.

. Loss of control of company data.

The stated advantages and disadvantages tend to confirm the cited advantages and disadvantages of distributed processing outlined in Chapter II.

VI

INCORPORATING THE PERSONAL COMPUTER

There are two major aspects of the problem presented in this thesis. The first aspect is: why bother changing the existing system? What overpowering reasons would dictate changing a working system? The second aspect is this: if change is needed, can it be accomplished? This second aspect, the doing, can be further subdivided into hardware and software considerations.

In Chapter II, the reasons for the change were discussed; mainly, to distribute the development of embedded systems down to the desktop level. Distributing the processing posed the objectives of performance improvement, lower costs, data sharing, reliability and flexibility.

In Chapter III, the hardware considerations were discussed, and a hardware model was presented. The hardware model consisted of the use of existing facilities to the greatest extent possible with an Ethernet-based local area network and DEC Rainbow 100 personal computers.

In Chapters IV and V, software considerations were discussed; specifically, what software tools are available and which tools receive the heaviest usage at Hughes Aircraft. The primary software tools that must be supported are the following: a compatible communications package; full screen editors; embedded software development tools comparable to AIDES and SDDL; FORTRAN, JOVIAL and CMS2/MTASS compilers; and system utilities for debugging and linking programs.

Accordingly, to incorporate the personal computer into the development of embedded systems, this chapter will attempt to answer the following questions:

A. Does the model fulfill the distributed processing objectives as they apply to Hughes Aircraft?

B. Does the model support the primary tools for embedded system development?

C. Does the model pose any major problems?

D. Does the model add anything to the tools for embedded systems development?

Distributed Processing Objectives

Performance improvement is directed toward system response time to the programmer's requests. The model will provide better response time since the PC is dedicated solely to the requirements of a single programmer. In addition, there is no contention for assets as in the existing system.

The model would not lower equipment costs because the PC's do cost more than the VT100 terminals, the local area network equipment is needed, and the existing communications facilities must remain in place.

Data sharing is not a heavy requirement in embedded systems development with the exception of transferring files to word processing, which is discussed in a subsequent paragraph. The amount of data shared between programmers is minimal under the existing system, so whether or not the model would improve data sharing is immaterial.

System reliablity would improve with the PC-based model.

Key component failure, such as the VAX, the communications lines or the multiplexor, would not cause a processing halt.

Flexibility would also increase. Incorporation of the DECMATE II word processors would give the system and programmers access to the installed letter-quality printers while retaining access to existing facilities.

Primary Software Tools

Communications in the PC world are still fairly well limited to terminal emulation, although both interactive and batch file transfer capabilities exist. A Rainbow 100 communications package, the poly-XFER CP/M communications software package, provides terminal emulation capabilities for file transfer between a variety of operating systems and information networks. The software package consists of four main components: poly-TRM, a terminal emulator for interactive communications and transmission of ASCII text files; XFR and HST, paired components designed for secure transmission and format conversion of files between the PC and other operating systems; and SWITCH, the fourth component, which permits remote operation of the PC. The application can be used to communicate with other CP/M hosts and with VAX hosts running VAX/VMS, PDP-11 hosts running RSTS/E, RSX-11M and RSX-11M-PLUS software. Another package, PHONELINK, allows batch mode communications without operator intervention.

An editor, ELECTRIC BLACKBOARD, is a multi-window full screen text editor for the PC. With the editor, the Rainbow 100 screen is divided into three smaller windows. These smaller windows are individual workspace viewing areas for each separate file that is being edited. The editor provides viewing, analysis and text editing for as many as ten different documents. The editor permits editing of text lines up to 254 characters in length, and a user can load, save, erase, move text, replace strings, and perform any editing function in any window. Windows are independent of each other with their own tab settings, page sizes, and cursor positions. This editor meets, and in some cases, exceeds the capabilities of the existing VAX editors.

There are no existing software development tools available for the Rainbow 100 PC comparable to SDDL and AIDES. The SDDL processor in use at Hughes Aircraft is PASCAL based and could be rehosted; however, it does not exist as of now. An AIDES-like tool is non-existent. Even in the terminal emulation mode, the PC cannot support the graphics necessary to run AIDES on the VAX. Given that AIDES ranked third in employee usage, lack of an AIDES-like tool is a serious drawback in the PC.

The Rainbow 100 has two FORTRAN compilers available, one of which supports the full 1966 ANSI standard with extensions, and a second with a number of enhancements to the ANSI standard. There are no JOVIAL or CMS-2/MTASS compilers available, although source code could be generated on the Rainbow for compiling on the VAX.

A programmer's utility package is available for library maintenance and linking relocatable object files. In addition, a symbolic instruction debugger is available which permits realtime breakpoint, full-monitored execution, symbolic disassembly, assembly and memory display as well as fill functions.

Major Problems

The single major problem with a PC-based system is data security. Although the model is electronically neither more nor less secure than the existing system, the data storage medium is changed from the VAX, with its attendant larger size, disk packs and 12-inch tapes to the smaller PC and the easily concealed 5 1/4-inch diskette. It is apparent that what the Xerox machine did to document security, the PC can do to data security.

A second problem with a PC-based system is found in the DECMATE II office system. Marbach felt that basing the office system on the "tried and true" PDP-8 made good sense. The counter to Marbach's opinion is that DEC may be trying to sell old technology with a redesigned cabinet. Although not applicable to this paper, basing the Professional-series on the PDP-11 falls into the same category. Since 1980, there has been a dramatic improvement in the capabilities of the hardware components. The Z80, 8088 and 68000 microprocessors, to name a few, have made dramatic improvements in the ability of the hardware to respond to user requirements. With DEC reissuing the PDP-8 and PDP-11 processors as new machines, either the processors were ahead of their time or DEC had no new development to offer. Some future researcher may want to investigate whether or not this is a real problem.

Added Tools

The majority of the existing software tools available for PC's are oriented toward the business user. Accordingly, the only two tools that would be added to the repertoire of tools presently available to the programmers at Hughes Aircraft are file transfer to word processing and administrative support.

File transfer between the document developers, who are the programmers in this case, and the formatters, who are the word processors, is much improved with the distributed model. The rough document can be developed by the programmer and then electronically transferred to the word processor for final editing and formatting.

Within the present Hughes Aircraft environment, most administrative functions are still handled manually. Data entry from the PC can replace the traditional manual keying of data from standard office forms such as expense accounts, purchase orders, or time card forms. The entry form can be called up on the screen and the blanks filled in. A software application package can take care of validation, error correction, editing, etc. A touch of the button sends the filled-in information to the host at the appropriate level for processing. Hughes Aircraft government contract work makes timekeeping critically important.

VII

CONCLUSION

To restate the first part of the hypothesis:

A. It is technically feasible to replace the intelligent terminals with personal computers.

This hypothesis is true. Technology has been identified in the PC-based model that can replace the VT100-based system with a distributed network.

To restate the second hypothesis:

B. A PC-based distributed local network would improve performance over the existing system for developing embedded systems without degrading the existing software engineering system.

The PC-based model will improve programmer performance, improve system reliability and improve system flexibility, which makes the first part of the hypothesis true. Adding the capability of transferring files electronically from the programmer to word processing is a sharp improvement over the existing manual transfer of written or printed material.

Finally, the PC-based model would also support all the tools presently available to the Hughes Aircraft programmer with one glaring exception. Implementation of the PC-based model would cause the loss of AIDES as a software development tool. Since AIDES is the third highest ranked tool in use at Hughes Aircraft, this factor makes the hypothesis false.

The final conclusion is that, although it is technically feasible to replace the intelligent terminals with PC's, all the software needed is not available to support embedded systems development. Accordingly, the PC-based system is not viable at this point in time.

Future Research

An aspect of a PC-based system which would provide an enormous assistance to the programmer at Hughes Aircraft was brought to light researching this paper; namely, on-line data base inquiry.

Data and document assets have accumulated all over the company. There are a few in the magnetic tape library, some in filing cabinets, more in the division, group or corporate libraries, and much more in people's heads. This internal data is in a variety of different storage and handling media including electronic, microform and hard copy.

External data sources, such as Source and Dialog, are also proliferating in media, form and content. Again, data and document assets are accumulating, but the average programmer does not have access to the information. While researching this paper, it was realized that the total information resources needed for effective software development are available, but they are disorganized, fragmented and compartmentalized. A future reseacher would benefit programmers immensely by developing a system to permit ready access to this data.

Other areas for future research include:

1. Is a personal computer available which can interface with the VAX and provide a tool comparable to AIDES?

2. Can an AIDES-like tool be developed for a PC?

3. Can SDDL be rehosted to a PC?

4. Are there any software developers specializing in software engineering tools and, if they exist, what are they marketing?

MEMORANDUM

From: D. Schultz

To: Hughes San Diego Employees

The attachment to this memorandum is a list of some of the tools available at Hughes for system development. I am trying to determine what tools are used and how often they are used. Since the best way to get the information is to ask the people who know, would you please take a few minutes and fill out the information requested. The survey can either be dropped in my mail slot or dropped off in my office (N-120).

How often do you use:

	Often	Occasionally	Rarely	Never
Compilers:				
ADA	________	________	________	________
BASIC	________	________	________	________
C	________	________	________	________
CMS-2	________	________	________	________
FORTRAN	________	________	________	________
INFORT	________	________	________	________
JOVIAL	________	________	________	________
MACRO	________	________	________	________
MTASS	________	________	________	________
PASCAL	________	________	________	________
RATFOR	________	________	________	________
Configuration Control:				
CCARS	________	________	________	________
Debugging:				
DEBUG	________	________	________	________
DUMP	________	________	________	________
PATCH	________	________	________	________
SIMULATOR	________	________	________	________
Documentation:				
RUNOFF	________	________	________	________
TECO	________	________	________	________
Editing:				
EDIT/EDT	________	________	________	________
EDIT/EVT	________	________	________	________
EDIT/SOS	________	________	________	________
EDIT/FDL	________	________	________	________
EDIT/SLP	________	________	________	________
EDIT/SUM	________	________	________	________
Management Information Report Generation (MIRG):				
MILESTONE REPORT	________	________	________	________
BARCHART SCHEDULE	________	________	________	________
STATISTICAL SUMMARY	________	________	________	________
System Development:				
AIDES	________	________	________	________
CMS	________	________	________	________
LIBRARY	________	________	________	________
LINK	________	________	________	________
RMS	________	________	________	________
RTL	________	________	________	________
SDDL	________	________	________	________

COLEMAN COLLEGE

Dear Sir:

The advent of the personal computer is having a profound affect on all areas of data processing. One area of primary importance in the San Diego area is the development of embedded systems. The question is then, does the personal computer have a place in the development of embedded systems? In an attempt to answer this question, I am gathering information on a better way to develop embedded systems using personal computers. As one of the prime players in the development of embedded systems, would you take a few minutes and answer the attached questionaire. The results of the questionaire will be used to identify the niche in software development where the personal computer belongs.

When you have finished the questionaire, please return it in the enclosed stamped, self-addressed envelope. The results of this survey wil be made available to you if you desire.

Yours truly

Darrell Schultz

I. The following is a list of software tools used to develop embedded systems. If the tool is available in your company, would you mark one column for how often it is used and one column for the host system.

	Frequency of Use			Host System	
	Often	Occasi- onally	Rarely	Personal Computer	Main- frame
Design Language	______	______	______	______	______
Compiling:					
ADA	______	______	______	______	______
ALGOL	______	______	______	______	______
APL	______	______	______	______	______
ASSEMBLY	______	______	______	______	______
BASIC	______	______	______	______	______
C	______	______	______	______	______
CMS-2	______	______	______	______	______
COBOL	______	______	______	______	______
FORTRAN	______	______	______	______	______
JOVIAL	______	______	______	______	______
LISP	______	______	______	______	______
MODULA	______	______	______	______	______
PASCAL	______	______	______	______	______
PL/1	______	______	______	______	______
RATFOR	______	______	______	______	______
Other: Please specify		____________		______	______
Editing:					
Full Screen	______	______	______	______	______
Line	______	______	______	______	______
Optimization	______	______	______	______	______
Data Flow Analysis	______	______	______	______	______
Complexity Measurement	______	______	______	______	______
Path Flow Tracing	______	______	______	______	______
Breakpoint Control	______	______	______	______	______
Logic Flow Tracing	______	______	______	______	______
Data Flow Tracing	______	______	______	______	______
Simulators	______	______	______	______	______
Symbolic Execution	______	______	______	______	______
Configuration Management	______	______	______	______	______
Project Management	______	______	______	______	______

	Frequency of Use			Host System	
	Often	Occasi-onally	Rarely	Personal Computer	Main-frame
Database Management	____	____	____	____	____
Change Control	____	____	____	____	____
Test Data Management	____	____	____	____	____
Files Management	____	____	____	____	____
Library Management	____	____	____	____	____
Documentation Management	____	____	____	____	____
Cost Estimation	____	____	____	____	____
Scheduling	____	____	____	____	____
Graphics:					
Flow Charts	____	____	____	____	____
HIPO Charts	____	____	____	____	____
Bar Charts	____	____	____	____	____
Milestone Charts	____	____	____	____	____
Structure Charts	____	____	____	____	____
Document Generation	____	____	____	____	____
Report Generation	____	____	____	____	____

II. If you develop embedded systems on a mainframe, please answer the following questions:

A. What type(s) of mainframe(s) do you use (e.g. VAX 11/780, 4341)?

AMDAHL	________	HONEYWELL	________
BURROUGHS	________	IBM	________
CDC	________	NCR	________
DATA GENERAL	________	PERKIN-ELMER	________
DEC	________	UNIVAC	________
GOULD SEL	________	OTHER	________

B. Do the systems you are developing require fragmenting to run on your mainframe?

Yes _____ No _____

C. Do your programmers lose any productive time due to the mainframe not being available (e.g. system down, in queue) ?

Never _____ Rarely _____ Occasionally _____ Often _____

III. If personal computers (PCs) are used for any aspect of embedded system development, please the answer the following:

A. What is your company's attitude towards the use of PCs?

Encourage use ____ Discourage use ____ No policy ____

B. Are the PCs company owned?

Yes _____ No _____

C. Are the PCs connected in a local network configuration?

Yes _____ No _____

D. Were any of the software development tools developed by your company ?

Yes _____ No ______

E. Does your company support any of the PC software packages (e.g. MULTIPLAN, WORDSTAR) ?

Yes _____ No _____

F. Do the PCs lessen control of company data?

Yes _____ No _____

G. Do the PCs cause greater configuration control problems?

Yes _____ No _____

IV. Generally:

A. What do you feel are the advantages of PCs in software development?

____

____

____

____

____

____

____

____

____

B. What are the disadvantages?

____

____

____

____

____

____

____

____

If you would like a copy of the results of this survey, please fill in the following:

Name ___________________________

Company ________________________

Address ________________________

City, State _____________________

BIBLIOGRAPHY

1. Houghton, Raymond C. JR.,"Software Development Tools: A Profile" Computer - IEEE Computer Society, May 1983 pp. 63-70

2. Marbach, Carl B., "Digital's New Personal Computers" The DEC Professional, September 1983 pp. 8-18

3. [Sprunk, W.] Coleman College, Distributed Processing and Databases, Undated, Session 9

4. Tanenbaum. Computer Networks New York: Prentice-Hall, Inc. 1981

5. Walsh, Myles E. Data Base & Data Communications Systems: A Guide For Managers Reston, Virginia: Reston Publishing Co, Inc. 1983

6. "Microcomputer Networks" Computerworld - On Communications, September 1983, pp. 99-105

7. Hughes Aircraft Company. Software Development System Manual, Volume 1 - 13 By permission.

Note: These publications contain internal and proprietary information and cannot be released as an open document.