2.1.4.2. Progress Highlights

In this section we summarize highlights of SUMEX-AIM resource activities over the past 4 years, focusing on the resource nucleus.

- We have continued to recruit new user projects and collaborators to explore further biomedical areas for applying AI. A number of these projects are built around the communications network facilities we have assembled, bringing together medical and computer science collaborators from remote institutions and making their research programs available to still other remote users. At the same time we have encouraged older mature projects to build their own computing environments thereby freeing up SUMEX resources for newer projects. Nine projects now operate on their own facilities, including three that have become BRTP resources in their own right. Nine projects in the community have completed their research goals and their staffs have moved on to new areas.

- SUMEX user projects have made good progress in developing and disseminating effective consultative computer programs for biomedical research. These performance programs provide expertise in analytical biochemical analyses and syntheses, clinical diagnosis and decision-making, molecular biology, and various kinds of cognitive and affective psychological modeling. We have worked hard to meet their needs and are grateful for their expressed appreciation (see Section 6).

- We have made significant strategic improvements to the SUMEX-AIM computing environment in order to optimize computing support for the community. These developed in ways somewhat different from the initially projected plan. The DEC VAX computer did not prove to be an effective machine for running Lisp [45], while Lisp workstations have in fact become available from a number of vendors as tentatively expected at the time of our proposal (first Xerox, then Symbolics and LMI, and more recently Hewlett-Packard and Texas Instruments). Thus, rather than augmenting our mainframe resources with the purchase of large address space VAX's, we upgraded the KI-TENEX system to a DEC 2060 and at the same time, began moving aggressively toward a Lisp workstation-based research environment, with the approval of an ad hoc site visit group. We did secure VAX capabilities for our community by means of access to an 11/780 purchased under DARPA funding. We made an initial purchase of Xerox Dolphins with NIH funding and subsequently added more Xerox and Symbolics machines with NIH and DARPA funding and with industrial gifts. Because of the broad mix of research in the SUMEX-AIM community, no single workstation vendor can meet our needs so we have undertaken long-term support of a heterogeneous computing environment, incorporating many types of machines linked through multiprotocol Ethernet facilities.

- We have continued the dissemination of SUMEX-AIM technology through various media. We have distributed various AI software tools to many research laboratories, including over 200 combined copies of the GENET, EMYCIN, AGE, MRS, SACON, GLISP, and BB-1 systems. Several of our software systems have been adapted as commercial AI tools such as the Teknowledge S.1 and M.1 systems derived from EMYCIN, the Texas Instruments Personal Consultant system derived from EMYCIN, and the IntelliCorp KEE system derived from UNITS. We have also prepared video tapes of some of our research projects including ONCOCIN and an overview tape of Knowledge Systems Laboratory work.

- Our group has continued to publish actively on the results of our research including more than 45 research papers per year in the AI literature and a dozen books in the past 5 years on various aspects of SUMEX-AIM AI research (see page 109). These books have included the three-volume set of the *Handbook of Artificial Intelligence*, edited by Barr, Cohen, and Feigenbaum; a book on *Readings in Medical Artificial Intelligence: The First Decade* by Clancey and Shortliffe; and a book on *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project* by Buchanan and Shortliffe.

- We completed the GENET project, begun in 1980 as a collaboration between the MOLGEN investigators and SUMEX, to make a set of DNA sequence analysis computing tools available to a national community of molecular biologists. This was an experiment in using a SUMEX-like resource to disseminate sophisticated software tools to a computer-naive community and proved extremely successful. GENET served over 300 molecular biologists before being phased out in early 1983. Subsequently, a new resource called BIONET has been funded by NIH at IntelliCorp to provide routine service of the type pioneered by SUMEX/GENET.

- A program in Medical Information Sciences was begun at Stanford in 1983 under Professor Shortliffe as Director. A group of faculty from the Medical School and the Computer Science Department argued that research in medical computing has historically been constrained by a lack of talented individuals who have a solid footing in both the medical and computer science fields. The specialized curriculum offered by the new program is intended to overcome the limitations of previous training options. It focusses on the development of a new generation of researchers with a commitment to developing new knowledge about optimal methods for developing practical computer-based solutions to biomedical needs. The feasibility of this program resulted in large part from the prior work and research computing environment provided by the SUMEX-AIM resource. Over 20 PhD and MS trainees will be enrolled in the fall of 1985. It has been awarded post-doctoral training support from the National Library of Medicine, received an equipment gift from Hewlett-Packard, and has received additional industrial and foundation grants for student support.

- We made significant progress in core AI research. In the area of knowledge representation, work was done on the representation of explicit strategy knowledge, temporal knowledge, causal knowledge, and knowledge in logic-based systems. In the area of architectures and control, we worked on a new implementation of a blackboard architecture with explicit control knowledge. Under knowledge acquisition studies, three PhD theses were completed covering experiments in learning by induction, by analogy, and learning from partial theories. In the area of knowledge utilization, results include work on reasoning with uncertainty and using counterfactual conditionals. We continued work on a number of existing tools for expert systems and on building new ones such as the BB1 system. And finally, significant work was done on the inference of user models, skeletal planning, defining a taxonomy of diagnostic methods, and reasoning with causal models.

- We have continued the core development of the SUMEX facility hardware, software, and networking systems to enhance the facilities available to researchers. Much of this work has centered on the effective integration of distributed computing resources in the form of mainframes, workstations, and servers. Network gateways and terminal interface machines based on

MC-68000 microprocessors were developed to link our environment together and are now the standard system used in the campus-wide Stanford University network. We developed a gateway interface between Apple equipment (e.g., the Macintosh and Lisa) and EtherNet hosts that is now in wide use at universities around the country. We have developed many other software packages to enhance the computing environments of the Lisp workstations and to link them to other hosts and servers on our networks.

The following sections then give more detail about SUMEX-AIM core resource activities since the last grant award.

2.1.4.3. Resource Equipment Details

The SUMEX-AIM core facility, started in March 1974, was built around a Digital Equipment Corporation (DEC) KI-10 computer and the TENEX operating system which was extended locally to support a dual processor configuration. Because of the operational load on the KI-10's, in the late 1970's, we had added a small DEC 2020 system (see Figure 2) to support more dedicated testing of systems like ONCOCIN and Caduceus and for community demos. This facility provided a superb base for the AI mission of SUMEX-AIM through 1982. Its interactive computing environment, its AI program development tools, and its network and interpersonal communication media were unsurpassed in other machine environments. Biomedical scientists found SUMEX easy to use in exploring applications of developing artificial intelligence programs for their own work and in stimulating more effective scientific exchanges with colleagues across the country. Coupled through wide-reaching network facilities, these tools also give us access to a large computer science research community, including active artificial intelligence and system development research groups.

The Heterogeneous Computing Environment

In the renewal for the current grant period, both an augmentation of the central resource in terms of address space and capacity and exploratory work with Lisp workstations were planned. The Initial Review Group recognized in their special study section report the importance of optimizing the timing of our planned hardware acquisitions to coordinate community needs with the availability of important technological developments in vendor-supported systems. They recommended in their report that we be allowed considerable flexibility as to phasing of equipment purchases within the 5-year renewal period.

We had initially planned to purchase a large VAX in 1981 and later, our first Lisp workstations. However, we speeded our push toward workstations for several reasons. The state of VAX Lisp implementations and projections of their performance were very discouraging (a study of the VAX InterLisp implementation was done at the time as documented in [45]). And the first Xerox InterLisp Dolphin workstations were available for delivery after the summer of 1981. These machines were the prototypes on which research toward adapting expert AI systems for the interactive workstation environment could begin. So, we purchased 5 Dolphins for the fall of 1981 and, in order not to delay non-Lisp SUMEX-AIM work involving VAX machines, we were able to arrange shared access to a VAX 11/780 funded by ARPA to support Heuristic Programming Project research. One of the Dolphins we purchased was loaned for several years to the Rutgers Computers in Biomedicine resource for experimental work.

We continued to evaluate strategies and alternatives for planned system configuration development. In particular, we had a chance to gain experience with the Dolphin InterLisp machines and the shared VAX, reassess the role of the dual KI-TENEX system, and reach a consensus about what the long term configuration of the SUMEX-AIM facility should be. This was validated by an ad hoc study section review in 1982. In summary, it was decided that the best resource configuration for the coming decade would be a shared central machine coupled through a high-performance network to growing clusters of personal workstations. The central machine should be an extended addressing TOPS-20 machine and the workstations will be chosen from the viable products available and scheduled for announcement.

The concept of the individual workstation, especially with the high-bandwidth graphics interface, proved ideal. Both program development tools and facilities for expert system user interactions were substantially improved over what is possible with a central time-shared system. The main shortcomings of these systems were their processing speed and cost, but the prospect of other workstations to be available from Xerox, Symbolics, LMI, HP, and others reassured us that these were the right choices for AI system in the long term. Still, at the time, it was not possible to equip very much of the SUMEX-AIM community with individual workstations.

Upgrade of the KI-10's to a 2060

Meanwhile, on the mainframe front, given the continued need for a central machine, the poor Lisp performance of the VAX, and the increasingly untenable difficulties in maintaining the KI-TENEX system, we decided it is time to retire the KI-10's and upgrade them to the then (1982) more modern DEC 2060 TOPS-20 system. This would free our systems staff to concentrate on more productive development efforts for the community such as work related to professional workstations and compatible Lisp support. The 2060 had a processing capacity of 2-3 times that of the dual KI-TENEX system, badly needed for our community, and it was more compact, reliable, and maintainable. Pending the arrival of more cost-effective and generally-available Lisp workstations, this would allow us to continue support for the SUMEX-AIM community at large and to provide facilities for new AI efforts.

In late 1982, we implemented the upgrade. The purchase price of the DECsystem 2060 reflected a substantial price reduction based on an external research grant from Digital Equipment Corporation to the Heuristic Programming Project in exchange for access by DEC to the AI software systems and knowledge-based systems expertise developed by the HPP. The remainder of the system was funded jointly by NIH and DARPA. The system configuration is shown in Figure 1. Of course, the transfer of service required a substantial investment of hardware engineering effort as all of the local line and network connections had to be changed over. This was all effected invisibly to the user community by running the old KI-TENEX and the new 2060 systems in parallel for more than a month.

Using DARPA funding, we also made some upgrades to the shared VAX 11/780 which was initially purchased by ARPA for HPP research as well as work in network graphics and VLSI design. The configuration of this machine is shown in Figure 3. In 1983, we augmented the machine by adding 2 Mbytes of memory and expanding the file system with a DEC RP07 disk drive (512 Mbytes). Approximately 60% of the machine is allocated for HPP and SUMEX use.

The overall facility model then became the central shared 2060, 2020, and VAX 11/780 systems surrounded with growing numbers of workstations and intercoupled by a local area network.

Additional Workstations

After the purchase of the 5 experimental Dolphin workstations, much work went into their development by Xerox, based on feedback and interactions with groups such as ours using them for AI applications. Performance of the Dolphins improved substantially based largely on improved microcoding of frequently used primitives and facilities. The initial optimizations of the Dolphin microcode were based on work at Xerox observing their own programs running. When the Dolphin was exposed to other AI systems such as ours, it became clear that additional improvements were necessary and were implemented, including enhanced performance for CONS operations, function calls, disk management, garbage collection, and other areas. Improvements in individual areas of performance ranged from factors of 2 to 10.

By 1983, other contenders were entering the Lisp workstation market in addition to Xerox. Because work in the HPP and the SUMEX-AIM community draws heavily on both Interlisp and the derivatives of MIT's MacLisp, we broadened our workstation experiments into both areas.

With NIH funding in 1983, we purchased 6 Xerox 1108 workstations (Dandelions) and in 1984, 3 Xerox 1109's (DandeTigers). With DARPA funding we purchased 2 Xerox 1108's and 1 1132 (high-performance Dorado) in 1984. In early 1985, the ONCOCIN group received a grant from Xerox of 13 1108's and additional printing and file server equipment. These machines represent the second generation of Xerox Lisp workstations and include significantly higher performance and functionality.

With DARPA funding in 1983 we bought a Symbolics LM-2 running the ZetaLisp system. In 1984, we added 3 Symbolics 3600's and a 3670 and in early 1985, another 3670 -- all with DARPA funding. We are also planning the purchase of additional workstations in the near term with DARPA funding.

Local Area Network Server Hardware

Since the late 1970's, we have been developing a local, high-speed Ethernet environment to provide a flexible basis for planned facility developments and the interconnection of a heterogeneous hardware environment. Our development of Ethernet facilities has been guided by the goals of providing the most effective range of services for SUMEX community needs while remaining compatible with and able to contribute to and draw upon network developments by other groups, dating back to the early 3 Mbit/sec Ethernet given to Stanford and several other universities by Xerox. We now support both 3 and 10 Mbit/sec Ethernets (see Figure 5) running numerous protocols and extended geographically throughout the SUMEX-AIM and related Stanford research groups. This network is the "glue" that holds the rest of the computing environment together and consists of numerous servers such as gateways and servers for terminal access, file storage and retrieval, and laser printing.

In the early phases, a substantial amount of special hardware was developed by our group for network interfaces including a high-performance direct memory access interface for the dual KI-TENEX system and a serial phase decoded UNIBUS interface that are used on our DEC 2020, VAX's, and early PDP-11 gateways and TIP's. The KI Ethernet interface served well for a period until we upgraded the system to a 2060, at which time we installed the 2060 mass bus EtherNet interface designed and built by the Stanford Computer Science Department. Our KI-10 interface is still seeing service in connecting another KI-10 system (Institute for Mathematical Studies in the Social Sciences) to the net.

Hardware for Gateways and TIP's

As we evolved a more complex network topology and decided to compartmentalize the overall Stanford internet to avoid electrical interactions during development and to facilitate different administrative conventions for the use of the various networks, we developed gateways to couple subnetworks together. These first used PDP-11/05 hardware and then Motorola MC-68000 systems as they became available.

Similarly, we designed gateway between Apple equipment such as the new Macintosh terminal, that may play a role in our future virtual graphics work (see page 162), and EtherNet using a MC-68000 gateway and a locally-designed Apple Bus to Multibus EtherNet interface. This system incorporates an 8530 Zilog chip to communicate with the Apple Net and software to manage the protocol packaging.

We also developed a MC-68000 terminal interface processor (TIP) to provide terminal access to network hosts and facilities. It is basically a machine that has a number of terminal lines and a network interface and software to manage the establishment of connections for each line and the flow of characters between the terminal and host. It can handle up to 32 lines. Both of these systems are now widely used throughout the Stanford network.

File Server Hardware

The development of an EtherNet file server was an integral part of our council-approved equipment plan with further expansions approved for later years. With joint NIH and DARPA funding, we were able to take advantage of an exceptional offer by Digital Equipment Corporation, through their corporate external research sponsorship program to DARPA contractors (the HPP), to purchase two VAX 11/750 machines as the processor part of our file servers. In the initial file server configurations, we also bought Fujitsu Eagle 450 MByte disks and controllers (one each from Systems Industries and Emulex) with one 800/1600 BPI tape unit for long term archives, and one 300 Mbyte removable pack drive for cyclic backups.

Other Network Hardware

We have developed numerous local network connection systems that have taken advantage of existing cabling rather than invest in expensive trenching and recabling. For example, in The Heuristic Programing Project (HPP) move to 701 Welch road, a high-performance network link to other SUMEX and campus network facilities was essential. Several communication schemes for establishing a reliable and relatively fast link were considered, including microwave, infrared laser, direct ethernet (by trenching and placing a direct ethernet cable), telephone company T1 service and others. All of these would have involved high cost and so we developed a communication link using bare copper telephone pair already in place. The wire distance between the HPP Welch Road location and the SUMEX machine room in the Medical Center is approximately 2000 ft. Utilizing high capacity differential drivers and ultra high speed, high sensitivity receivers, a half-duplex transceiver was developed for plain copper twisted pair that achieved error-free transmission at 1.25 Mbits/sec in each direction, utilizing Manchester data encoding. This communication link has been in operation for well over a year now without any appreciable down time or noticeable error rate or data delays.

In addition to the normal continuous flow of maintenance problems, we have reconnected the very reliable line printer from the old KI-TENEX system to the 2060. This required substantial modification of the printer controller to adapt to the different 2060 bus signal standards. We have also installed lots of communications equipment, including dial-in and -out modems and laser printer connections.

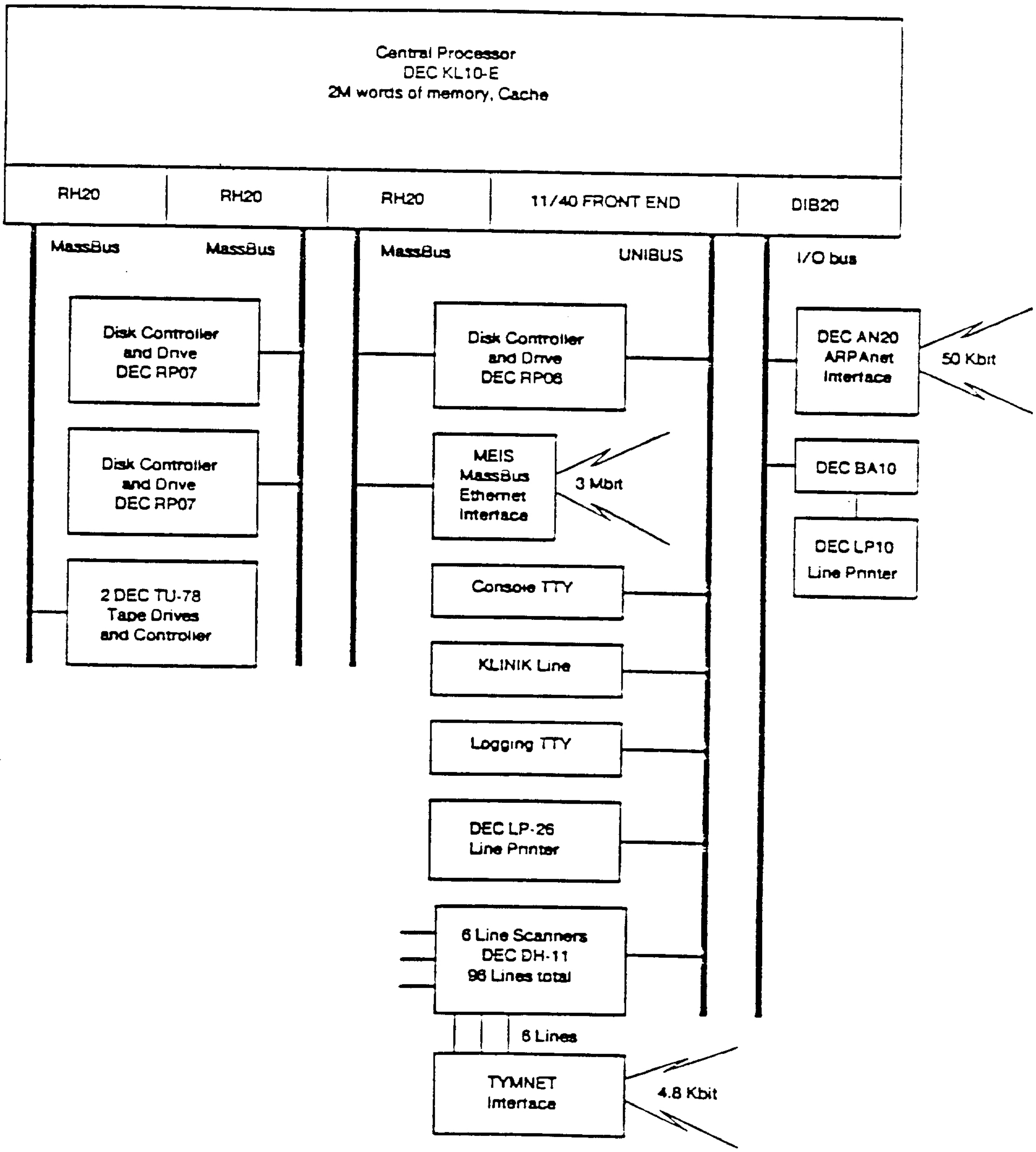

Figure 1: SUMEX-AIM DEC 2060 Configuration

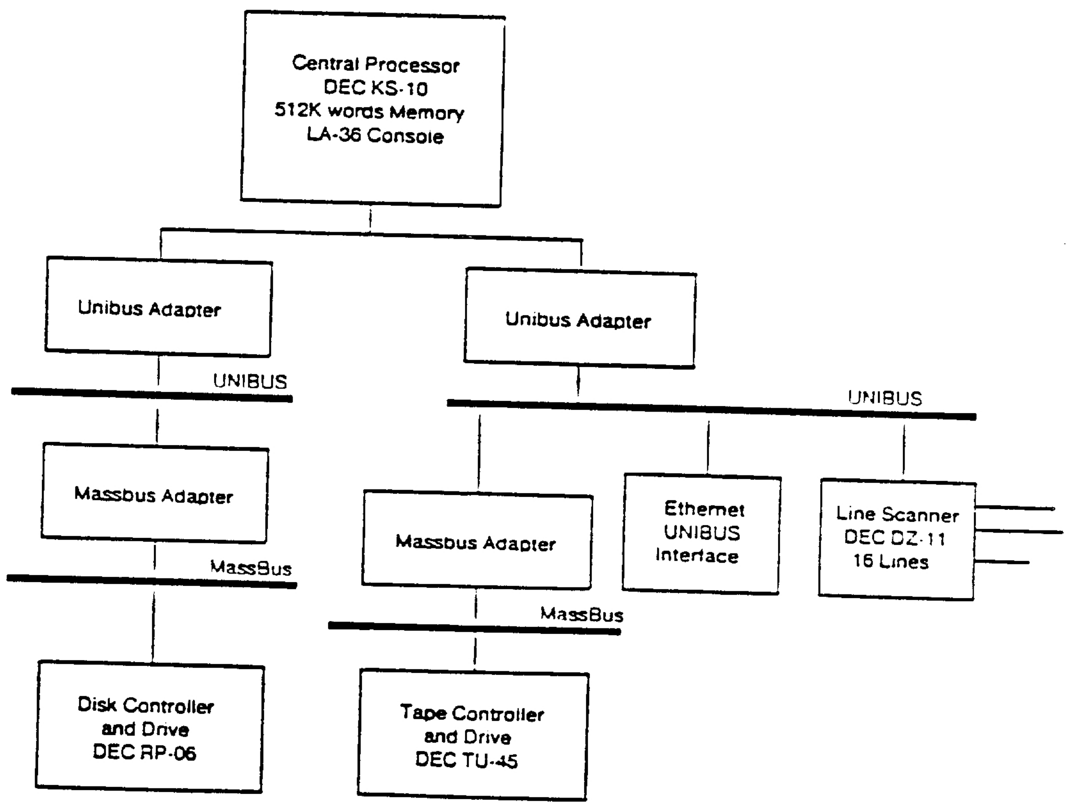

Figure 2: SUMEX-AIM DEC 2020 Configuration

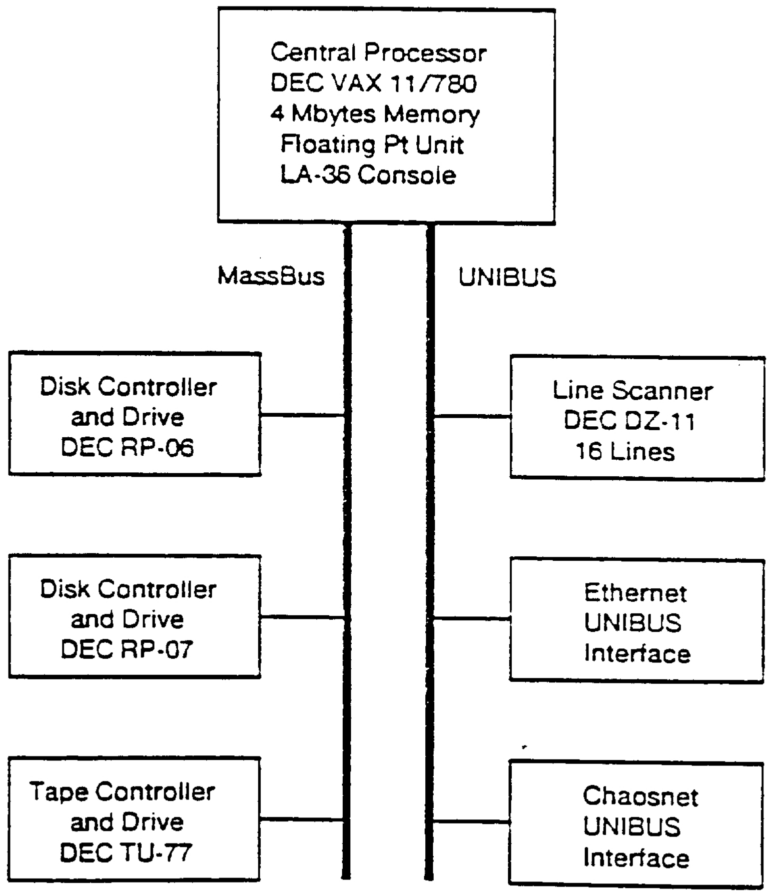

Figure 3: SUMEX-AIM Shared DEC VAX 11/780 Configuration

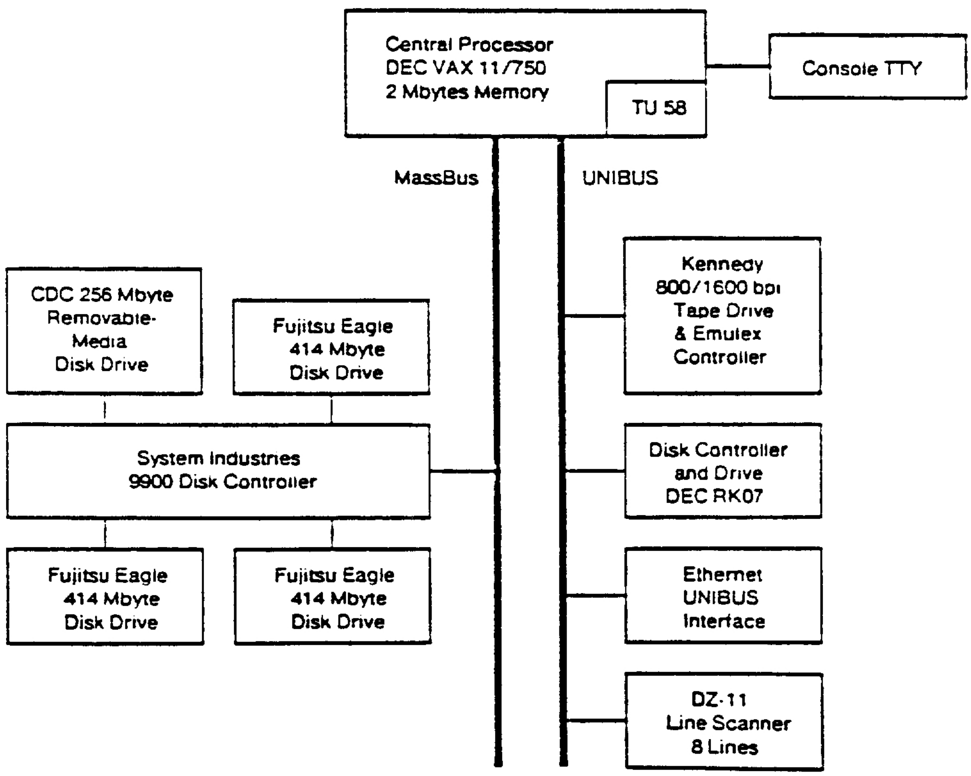

Figure 4: SUMEX-AIM File Server Configuration

2.1.4.4. Core System Development

Operating System Software

The various hardware elements of the SUMEX-AIM computing environment require the development and support of the operating systems that provide the interface between user software and the raw computing capacity. In addition to performance and relevance to AI research, much of our strategy for hardware selection has been based on being able to share development of the operating systems for our research among a large computer science community. This includes the mainframe systems (TOPS-20 and UNIX) and the workstation systems. Following are some highlights of recent system software developments.

TOPS-20 Development

The upgrade of the KI-TENEX system to the 2060 required a very large effort. Whereas the KI-TENEX system contained a great many local enhancements and adaptations, our goal was to run a TOPS-20 system that was broadly supported but which also tracked research developments outside of those motivated by vendor commercial interests. The most obvious choice for our immediate system peer community was the other 6 DEC 2060 sites at Stanford since we shared common internet problems and also had common goals in supporting research work rather than production computing. We also, of course, retained contact with the other ARPANET computer science systems. This course has constrained our own local developments by being part of a larger group of peers but the added problems of coordination have required fewer site-specific extensions and customizations at the operating system level.

Given this perspective, the following are specific areas of TOPS-20 system effort:

- In the conversion from TENEX, much planning and effort went into moving the file system, along with the pertinent user-specific directory information. In addition, we were able to preserve access to the vast magnetic tape library of archived and otherwise backed up files that had been created and saved since the inception of SUMEX. A TOPS-20 version of BSYS, a file archiving system, was imported from ISI as part of the effort to convert to the 2060. Numerous changes were made to make it compatible with the version of BSYS previously used at SUMEX. The LOOKUP program, used under TENEX, was converted to TOPS-20 use and made compatible with the new version of BSYS. We reviewed and updated appropriate documentation files in the HLP: and DOC: directories. And we identified and upgraded numerous system utility programs that utilized TENEX-dependent system calls.

- Using Tenex code previously developed at SUMEX as a base, we added new code to the TOPS-20 monitor to significantly enhance the user interface to the file system naming primitives. One addition was intercepting a ? typed by a user as part of a file name, then displaying for the user the valid file name alternatives matching the type-in up to that point, and finally returning to the original context, allowing the user to continue typing where he left off. Another addition was to generalize the logic involved in file name recognition in the case where more than one file matches what is typed in at the point where the request for recognition was given. The new logic looks ahead at the alternatives and fills out as much of the file name as possible, i.e. up to the point of ambiguity.

- Continued development of QANAL (formerly ANAL), a crash analysis program that has been under development since 1978. This program significantly eases the burden of analyzing the causes of system crashes due to both hardware and software problems. In addition, the accumulated outputs from QANAL allow for the detection of long term crash correlations to analyze infrequent problems.

- Track network protocol and service (e.g., file transfer and electronic mail) developments. We coordinated SUMEX's changes required to support the ARPANET-wide change from the old NCP protocols to the DOD IP/TCP protocols. This complex software required significant effort on our part because SUMEX-AIM has become a major communications crossroads and so exercises the network code very heavily. This has raised many problems of bugs and performance that we have worked to improve. We have played an active role in network discussion groups related to areas such as electronic mail, network designs, and protocols and had kept system tables for network host names and addresses, both local and over the ARPANET, up-to-date.

- Developed expanded file system support through multiple RP07 disk drive service. We were the first site to support more than one RP07 unit in a single structure.

- Implemented support for the old but superior LP10 printer from the KI-TENEX system. Even though DEC doesn't support this configuration, the LP10 has become our standard printer.

- Implemented subdirectory access to allow users full "owner" access to their subdirectories via the Access Control Job.

- Developed improved system allocation code, including the ability to withhold scheduler "windfall" from a given class or classes, with associated code in SKED% JSYS.

- Improved the efficiency of file backup and archive facilities by flagging directories with ARCHIVE and MIGRATE requests pending rather than searching through all directories serially.

- We have done substantial work on the TOPS-20 system Executive, the program that serves as the primary interface between users and the system. It provides commands to manipulate files, directories, and devices; control job and terminal parameter settings; observe job and system status; and execute public and private programs. The SUMEX EXEC is quite well developed at this stage but we have made several improvements. For example, we added a command line editor developed at the University of Texas and commands for the various laser printer spooling capabilities described later. There were also many more minor upgrades such as reading SYSTEM:LOGIN.CMD and SYSTEM:COMAND.CMD files on user login, account verification, enhancing various information commands, and improved directory and file system facilities to assist users in managing their files.

We have made numerous monitor bug and hardware problem repairs to provide for more reliable system operation and file integrity. Obvious bugs were removed long ago so those remaining are elusive and difficult to track down. We have also spent time keeping up-to-date with the latest monitor releases.

VAX 4.2 BSD UNIX Development

We run UNIX on our shared VAX 11/780 and on our 11/750 file servers. This system has been used pretty much as distributed by the University of California at Berkeley, except for local network support modifications. The local VAX user community is small so we have not expended much system effort beyond staying current with operating system releases and with useful UNIX community developments. The SUMEX VAX was the first site at Stanford to bring up the Berkeley 4.2 BSD distribution in October 1983. Since this was an early distribution, there were quite a number of bug fixes required; these were accomplished both through local effort and through monitoring the unix-wizards mailing list. After this kernel was running on the SUMEX machine, it was transported other sites and became the basis for the campus-wide UNIX 4.2 distribution.

To allow the UNIX network interface code to work in our Stanford subnet environment, we created a pseudo-network interface driver called 'sub0', that routed all output IP datagrams, based on their subnet numbers. This driver was done transparently, so that at system boot time, you could configure the machine for Stanford subnets, or for normal network routing. We also worked with other Stanford sites to install the Stanford PUP network drivers and servers back into 4.2 BSD (Berkeley does not support these).

Workstation System Development

Lisp workstations represent the major new direction for system development at SUMEX-AIM because these machines offer high performance Lisp engines, large address spaces required for sophisticated AI systems, flexible graphics interfaces for users, state-of-the-art program development and debugging tools, and a modularity that promises to be the vehicle for disseminating AI systems into user environments. We have accordingly invested a large part of our system effort in developing selected workstations and the related networking environments for effective use in the SUMEX-AIM community.

Xerox D-Machines

Much of the SUMEX-AIM community uses InterLisp and has moved naturally to the Xerox D-machines -- initially the Dolphin and then the Dandelion, Dandetiger, and Dorado. Much work has gone into hardware installation and networking support but we have also developed numerous software packages to help make the machines more effective for users and to ease our own problems in managing the distributed workstation environment.

In the transition to workstations as *computing* environments suitable for AI applications work, not just as programming environments, much system development remains to be done. One of the problems we have examined and plan to continue to exploring is that of building distributed expert systems. We are interested, for example, in separating the reasoning components and user interfaces and are designing a system with multiple processes which can run on a single or multiple workstations in order to independently develop, tune and evaluate the components. To facilitate this we have developed a prototype inter-process message passing interface which makes the topology of the system invisible to communicating processes, whether on one machine or several CPU's linked via the Ethernet.

Another of our interests is in exploring how to combine different software and/or hardware architectures in order to take advantage of the best features of each. One simple low level program that we built allows us to use Interlisp workstations to down load software into Mesa workstations in order to boot them using the Ethernet as an alternative to the hard or floppy disk drives. Along the same lines, we are exploring efficient ways to communicate high level descriptions of graphic data among differing media. We have developed a simple system which will take text formatting files and translate them into graphic window displays, defining active regions of the screen in the process. This facilitates the design of user interfaces using the familiar medium of text processing.

In our AI systems work, we have developed a low overhead object-oriented system which is designed to be flexible enough to model different object-oriented programming styles at the same time. It is also designed to facilitate a model of large knowledge bases which reside principally on file servers but whose components are loaded on demand. With this system, a minimal set of information about all the objects in a knowledge base is loaded upon opening. This information allows many simple inquires about the nature of objects and their relationships to be made without the main body of the object being resident. Only when non-trivial operations are performed are the contents of the object brought into core. This design is based on the belief that the size of knowledge bases will eventually grow to exceed the capacity of any given computer. However, most systems will generally only need a manageable subset of objects at runtime.

Other work we have done includes monitoring tools to examine static function calling hierarchy as well as view runtime executions graphically. We are also developing graphics interfaces to knowledge base construction and maintenance.

Some of the InterLisp software packages that have been written in the course of this work include:

ACFontCreate -- Reads a Xerox PARC font file in AC format into a lisp data structure

BaudRate -- Benchmarks baudrates by BINing through a file

DSys -- Monitors D machine usage on demand

GraphNet -- Derives topology of the PUP internet via net and gateway probes

HPColor -- Interlisp image stream implementation to drive H-P dgl graphics

Impress -- Interlisp image stream implementation to generate Impress print files

MakeStrike -- Writes out an Interlisp display font as a strike file

MLabel -- Generates mailing labels from a mailing list

RasterFontCreate -- Generates an Impress font of bitmap patches in arbitrary scale

ReadRSTFontFile -- Reads an Impress font file into a list data structure

RemoteTools -- Tools to manipulate a remote Interlisp using its systat process

RootPicture -- Reads a Press file bitmap into a lisp bitmap

RSTSample -- Creates an Impress sampler showing all characters of a font

SIL -- Reads and displays a SIL drawing file and optionally hardcopies it

SYSTAT -- a remote Eval server for Interlisp

Undither -- Compresses a previously dithered image into an AIS file

VDSDog -- Monitors array space usage to prevent crashing from lack thereof

WriteRSTFontFile -- generates an Impress font file from a special Lisp structure

ZDir -- TENEX-style directory lister for use with UNIX via Leaf server calls

DScribe -- A simple SCRIBE-to-display list parser/driver.

EtherBoot -- Provides microcode and program boot service for Xerox 8000's

GraphCalls -- Graphs the calling hierarchy of a lisp function and more

Hash -- Provide a machine independent hash file facility

EditBG -- A background/border texture editor.

FileLstW -- Menu-based interface to the file package.

MagnifyW -- A magnifying glass for bitmaps.

Message -- Multi-process/Multi-CPU message passing facility.

MultiW -- Links windows so that they move, surface, and close as a group

OZone -- An object-oriented programming system for Interlisp

Plotter -- Interlisp image stream to generate native-mode H-P plot files

Register -- Bundles menus into a coherent device for complex input

Region -- A utility to allow dissimilar activity in a single window.

Storage -- A utility to display Interlisp data type storage graphically.

Once a package has been developed and determined to be of general interest, we announce it over an electronic mail users list and make it available to other sites. In some cases, packages have such extensive utility that they are submitted as LispUsers packages for distribution by Xerox. This occurred in the case of Graphcalls, Hash, MultiW, and FileLstW, the latter submitted under the name Manager.

We have worked closely with many other sites, including the Center for Study of Language and Information at Stanford, the Stanford Campus Networking group, Rutgers University, Ohio State University, the University of Pittsburgh, Cornell, Maryland, and industrial research groups such as Xerox Palo Alto Research Center, SRI, Teknowledge, IntelliCorp, and Schlumberger-Doll Research. We have been the maintainers for the international electronic mail network of users for research D-machines, which have upwards of 300 readers, and the interchange of ideas and problems among this group has been of great service to all users.

Symbolics Lisp Machines

We have a growing community of Symbolics machines and users. Little development has gone into the tools for these systems yet because the small number of machines we have are concentrated in applications groups. We have actively supported the installation and maintenance of these systems, the installation of new software releases, and the integration of these systems with the rest of our networking environment. We were a beta test site for the Symbolics IP/TCP software.

Macintosh Workstations

In early 1984 Apple Computer released their new Macintosh and we were immediately interested in it as a possible low-cost display workstation to interface to our Lisp workstations and other hosts. In order to evaluate the Macintosh for this purpose, SUMEX received some early equipment and manuals through Stanford's participation in the Apple university consortium program. Like many groups trying to experiment with Macintosh software however, we found the Apple Lisa cross-development environment somewhat restrictive and hard to use and this was the only way to create Macintosh software at the time. So we built a UNIX-based cross-development environment on our VAX. It turns out, that this was the first C development environment available on the Macintosh when we released our software (via Arpanet FTP) in June of 1984. SUMacC (Stanford University Macintosh C) has been quite widely received, and is in use at well over a hundred sites throughout the US and in foreign countries. SUMACC integrated pieces of software from many groups, and was therefore something of a cooperative effort. We have openly distributed it to other users either through network FTP or a magnetic tape at distribution cost. Version 2.0 of the SUMACC system was released in November of 1984.

Among the many useful programs subsequently written with SUMACC were: (1) a Kermit program done at Harvard, (2) the Mac PSL (Portable Standard LISP) done at the University of Utah, and (3) an 'external file system' done by John Seamons of LucasFilm which allows the Macintosh to use an Ethernet host (such as UNIX) as a general network file server (see also page 95).

With the increased usage of Macintoshes in the SUMEX-AIM community, the need to be able to transfer files between them and TOPS-20 mainframes quickly arose. We therefore reimplemented the MACGet and MACPut file transfer utilities, previously developed for UNIX, for TOPS-20. These incorporated TOPS-20 style terminal handling and file system conventions. Both programs provide reliable (i.e., checksummed) transfer of either text or binary data, and are now gaining wide-spread use outside of SUMEX.

Virtual Workstation Graphics

Finally, we have done a number of experiments with the remote connection of bitmapped displays to hosts and workstations. Generally, the displays on Lisp machines are tethered through a high bandwidth cable to their processors. This limits the flexibility with which users can move from one Lisp machine to another (one must move physically to another machine) and loses the ability of researchers to work from home over telephone lines. A way of providing more flexible display to processor connection is to use a virtual graphics protocol, such as the V Kernel system developed by Lantz [37], that allows efficient communication of the contents to be displayed on a bitmapped screen. In an initial experiment, an Interlisp virtual graphics module was written to run on the DEC-2060 and drive the graphics engine of a Sun MicroSystems workstation over the Ethernet. This system allows applications running on the DEC-2060 to create views, and windows within those views on the remote workstation, and then using the Virtual Graphics Terminal Protocols, manipulate those views and windows. One can place text, draw objects such as points, lines, shaded rectangles, splines, and bitmaps in these screen areas. Local and remote editing of the graphics representation is also possible with a responsiveness close to that of a directly connected display.

Network Services

A highly important aspect of the SUMEX system is effective communication within our growing distributed computing environment and with remote users. In addition to the economic arguments for terminal access, networking offers other advantages for shared computing. These include improved inter-user communications, more effective software sharing, uniform user access to multiple machines and special purpose resources, convenient file transfers, more effective backup, and co-processing between remote machines. Networks are crucial for maintaining the collaborative scientific and software contacts within the SUMEX-AIM community.

Remote Networks

We continue our connection to TYMNET as the primary means for access to SUMEX-AIM from research groups around the country and abroad. Substantial work was required to transfer TYMNET service from the KI-TENEX system to the 2060 because the new system does not support the same memory-sharing interface we had for the KI-10's. There has been no significant change in user service or network performance though. Very limited facilities for file transfer exist and no improvements appear to be forthcoming soon. Services continue to be purchased jointly with the Rutgers Computers in Biomedicine resource to maximize our volume usage price break. We continue to have serious difficulties getting needed service from TYMNET for debugging network problems and users away from major cities have problems with echo response times.

We also continue our extremely advantageous connection to the Department of Defense's ARPANET, managed by the Defense Communications Agency (DCA). This connection has been possible because of the long-standing basic research effort in AI within the Knowledge Systems Laboratory that is funded by DARPA. Terminal access restrictions are in force so that only users affiliated with DoD-supported contractors may use TELNET facilities. ARPANET is the primary link between SUMEX and other machine resources such as Rutgers-AIM and the large AI computer science community supported by DARPA. Our early Honeywell IMP has been upgraded to a BBN C/30 IMP in preparation for the transition to the IP/TCP protocols. We are also investigating the installation of a link to the DARPA wideband satellite network to facilitate the rapid transfer of large amounts of data such as are involved with projects like our Concurrent Symbolic Computing Architectures project.

Local Area Networks

For many years now, we have been developing our local area networking systems to enhance the facilities available to researchers. Much of this work has centered on the effective integration of distributed computing resources in the form of mainframes, workstations, and servers. Network gateways and terminal interface processors (TIP's) were developed and extended to link our environment together and are now the standard system used in the campus-wide Stanford University network. We are developing gateways to interface other equipment as needed too (e.g., the Macintosh and Lisa). A diagram of our local area network system is shown in Figure 5 on 94 and the following summarizes our LAN-related development work.

MC-68000 Server Kernel -- Our early network gateways and TIP's were based on PDP-11 systems. But these soon became limiting in terms of speed, address space, and cost. With the introduction of the Motorola MC-68000 microprocessor and its integration into a compact, large-memory machine in the prototype SUN processor board developed in the Computer Systems Laboratory at Stanford, a much better vehicle was at hand. The net server software we developed for the PDP-11 included a kernel which handles hardware interfaces, core allocation, process scheduling, and low-level network protocol management. The 3 MBit/sec Ethernet PDP-11 based PUP kernel was translated and augmented for the MC-68000 CPU/SUN ethernet interface. This kernel then became the basis for the SUMEX gateway and TIP software which both have become the Stanford standard. As networking technology developed, the SUMEX kernel was extended to include 10 MBit/sec Ethernet drivers and to support 10 Mbit/sec PUP, XNS, and IP protocols. The main modification needed was the addition of a 10 MBit/sec Ethernet address resolution protocol module so that a 10 MBit/sec PUP host could discover its "soft" PUP address from a cooperating gateway on its local network.

Ethernet TIP -- Based on the new augmented MC-68000 kernel, the 3 Mbit/sec PDP-11 Ether TIP code was translated. This new TIP could handle increments of 8

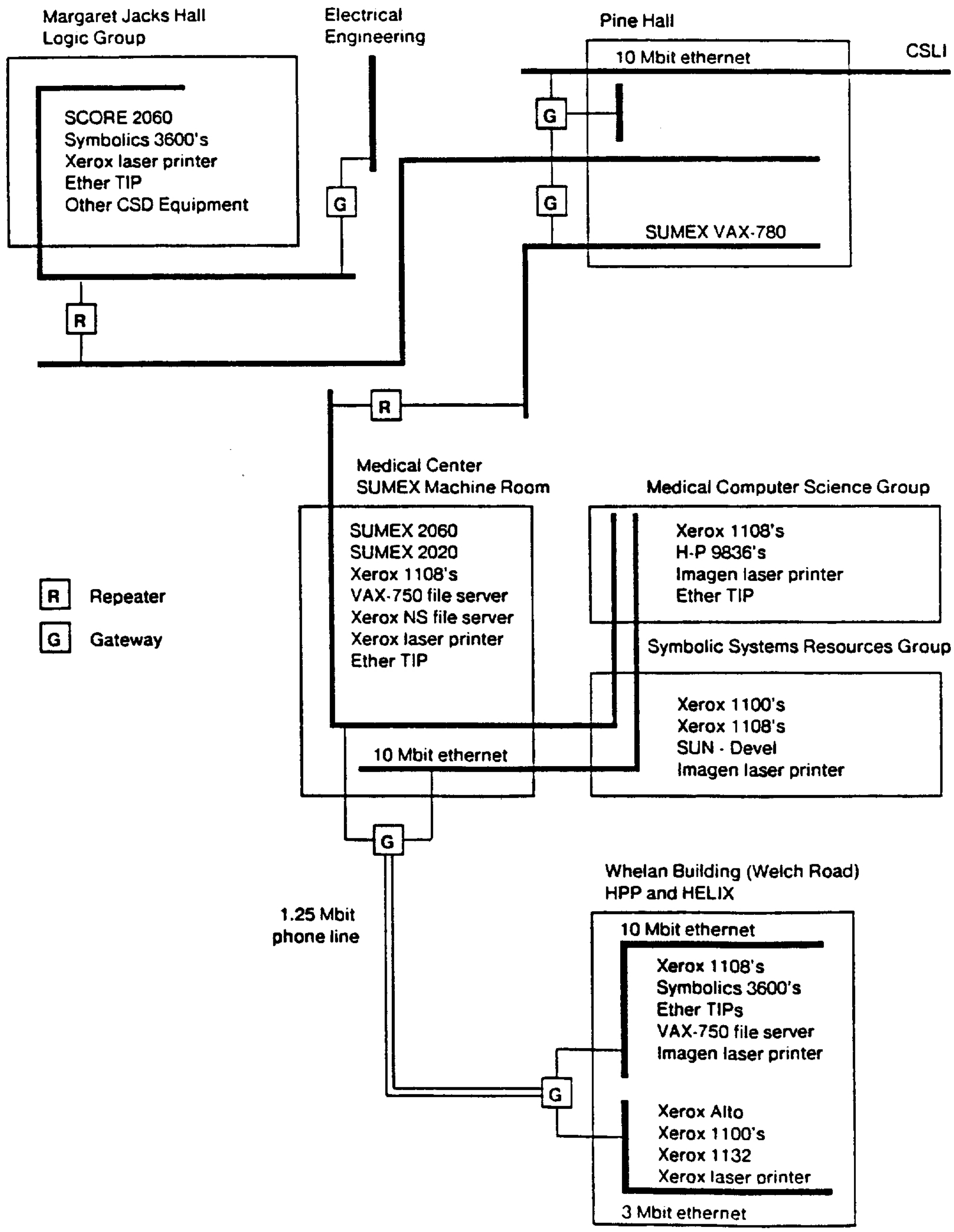

Figure 5: SUMEX-AIM EtherNet Configuration

lines up to 32 lines in a six slot backplane. With the advent of the newer 16 line DUART's developed in the Stanford Computer Science Department, 80 line TIP's have been built using this TIP code. This code is still running on several 3 Mbit/sec Ether TIP's at SUMEX. As 10 Mbit/sec networks were introduced, the TIP code was updated and adapted so that TIP's could run on either 3 MBit/sec or 10 MBit/sec Ethernets. There are now over 20 TIP's installed at Stanford using the SUMEX code and the number will increase substantially as the campus-wide local area network grows. The development of this software is essentially complete now with the recent addition of an improved user interface and facilities for inbound connections (such as for remote printers).

Ethernet Gateways -- Like the TIP systems, the PDP-11 gateway code was adapted to the MC-68000 hardware and extended to both 3 Mbit/sec and 10 Mbit/sec networks. Gateways can be configured to support up to four directly connected networks which may be either 3 MBit/sec or 10 MBit/sec. The gateway system was made "self-configuring" so that only one bootable gateway was needed. Network directory downloading and name/address lookup services were added. The routing algorithm was rewritten to minimize probe time for efficiency because of the continued growth of the number of subnetworks in the Stanford University network. The gateway now supports PUP and IP packet transport and XNS packet routing for both 10mb and 3mb networks is being completed. There are over twenty SUMEX gateways installed at Stanford and this number should double in the next year.

A special gateway configuration was required for the HPP move to Welch Road. Since the physical link was differentially driven 1.25 MBit/sec twisted pair cable, the network connections required two three-way gateways, one at either end, and special hardware to interface the serial lines with the ethernet interfaces. The required special hardware and software were built and the WR gateway has operated very effectively.

Apple Gateway Another special gateway, named SEAGATE, was developed to better integrate the Apple Macintosh into our Ethernet system. It links the Ethernet and Apple's AppleBus/AppleTalk network. This was completed and released in February 1985. Several internet sites, including some at Stanford, are currently constructing duplicate gateways. Also, several commercial firms are building a one board version of the gateway which should lower the cost to about $1000 per gateway. EFS, MAT, and AppleTalk Library are some sample Macintosh programs and UNIX daemons, that utilize SEAGATE. EFS is an external file system, written by John Seamons, and modified by us to work over AppleTalk. With EFS the Mac user sees his normal iconic view of the world. His UNIX directory appears as an icon and he can remotely execute and transfer files, simply by clicking on their icons. EFS is to the Mac as Leaf is to a LISP machine. The AppleTalk library is used by all of these programs to perform the ATP protocol (AppleTalk transaction protocol). This is the general protocol used to perform printing, file transfer, etc. with the Mac. The library allows a UNIX user-level process to perform this ATP protocol. Note that no kernel changes are required, since the ATP datagrams are imbedded in IP datagrams (UDP) by the SEAGATE. MAT is the Mac ATP Transfer program, a sample program that does file transfers with a UNIX host. It can also act as the framework for a Mac mail or print service.

Remote File Service -- In a distributed workstation environment, effective file access and transfer facilities between workstations and other hosts and servers are a must, especially to file servers like those we built around VAX 11/750 UNIX systems. Initial file service support used code written as a student project in the Stanford Computer Systems Laboratory. But as the number of workstations increased, service degraded and it became necessary to rewrite the PUP/BSP UNIX software package, and major portions of those programs dependent upon these protocols. This resulted in a 300% increase in throughput and stabilized the Lisp Machine to VAX 11/750 file service environment. At the same time we made major improvements to the UNIX Leaf service for XEROX D-machines. The earlier code, again a student systems project, had many bugs and inefficiencies and required a complete rewrite. In the new code, each Leaf connection was given a separate process to manage its Leaf resources, whereas previously, all users' Leaf requests were simply handled as a serial queue. This meant that every packet created a bottleneck for its successors. This work resulted in a much better Leaf service environment with considerable improvement in overall responsiveness and throughput.

Laser Printing Services

Since the first Xerox laser printers were developed in the mid-1970's, several companies have produced computer-driven systems, such as the Xerox Raven and the Imagen 8/300. These systems have become essential components of the work of the SUMEX-AIM community with applications ranging from scientific publications to hardcopy graphics output for ONCOCIN chemotherapy protocol patient charts. We have done much systems work to integrate laser printers into the SUMEX network environment so they would be routinely accessible from hosts and workstations alike.

We collaborated to develop an Ethernet interface for Imagen printers starting about January of 1984. We arranged to upgrade our Imprint-10 controller in exchange for the UNIX software needed to drive it from the network and were the first site to receive this controller in beta test stage. The UNIX software we developed made it possible to connect the printer to the new 4.2 BSD line printer spooler package using IP/TCP protocols. This was completed about March of 1984. After the UNIX implementation was complete, we developed the corresponding TOPS20 software to interface to this new printer and later, integrated it into the TOPS20 Galaxy spooler package. Other sites on campus and in the internet, began using the new printer and our spooling software as well.

We similarly developed and enhance the spooling system for the Dover and Alto-Raven laser printers and added a header page for Raven output to separate listings. And in addition to the device support for the printers to interface to the various mainframe hosts machines in our network, we also developed packages to allow Xerox D-machines and Symbolics 3600 machines to print to the networked laser printers.

On the SUMEX-AIM mainframe hosts, SCRIBE is the predominant document compilation system, but in the initial stages, it was essentially only used with the Xerox Dover printer or a daisywheel typewriter. In the succeeding years we integrated the Imagen Imprint-10 driver from Unilogic, brought up the Xerox Alto-Raven, and installed support for the new group of Imagen printers (the 8/300's), which are based on a Canon copier and are now the workhorse printing resources of the local community. We made numerous improvements in the printing fonts available to users, including a rework of Knuth's Computer Modern Roman fonts for a more contemporary look on the Imprint-10, creating a sans serif font family based on Computer Modern Roman, generating Helvetica and Times Roman font families from the Xerox sources used to generate the Dover fonts, and creating and improving many document types in use by the community.

General User Software

We have continued to assemble (develop where necessary) and maintain a broad range of user support software. These include such tools as language systems, statistics packages, DEC-supplied programs, text editors, text search programs, file space management programs, graphics support, a batch program execution monitor, text formatting and justification assistance, magnetic tape conversion aids, and user information/help assistance programs.

A particularly important area of user software for our community effort is a set of tools for inter-user communications. We have built up a group of programs to facilitate many aspects of communications including interpersonal electronic mail, a "bulletin board" system for various special interest groups to bridge the gap between private mail and formal system documents, and tools for terminal connections and file transfers between SUMEX and various external hosts. Examples of work on these sorts of programs have already been mentioned in earlier sections on operating systems and networking. A further gratifying example is the TTYFTP program, originally written at SUMEX as a system for file transfers usable over any circuit that appears as a terminal line to the operating system (hardline, dial-up, TYMNET, etc.) and incorporating appropriate control protocols and error checking. The design was derived from the DIALNET protocols developed at the Stanford AI Laboratory with extensions to allow both user and server modules to run as user processes without operating system changes. TTYFTP formed the basis for the KERMIT program that is now distributed by Columbia University and which is in very wide use for communications between personal computers and to mainframe hosts.

At SUMEX-AIM we are committed to importing rather than reinventing software where possible. As noted above, a number of the packages we have brought up are from outside groups. Many avenues exist for sharing between the system staff, various user projects, other facilities, and vendors. The availability of fast and convenient communication facilities coupling communities of computer facilities has made possible effective intergroup cooperation and decentralized maintenance of software packages. The many operating system and system software interest groups (e.g., TOPS-20, UNIX, D-Machines, network protocols, etc.) that have grown up by means of the ARPANET have been a good model for this kind of exchange. The other major advantage is that as a by-product of the constant communication about particular software, personal connections between staff members of the various sites develop. These connections serve to pass general information about software tools and to encourage the exchange of ideas among the sites and even vendors as appropriate to our research mission. We continue to import significant amounts of system software from other ARPANET sites, reciprocating with our own local developments. Interactions have included mutual backup support, experience with various hardware configurations, experience with new types of computers and operating systems, designs for local networks, operating system enhancements, utility or language software, and user project collaborations. We have assisted groups that have interacted with SUMEX user projects get access to software available in our community (for more details, see the section on Dissemination on page 109).

Operations and Support

The diverse computing environment that SUMEX-AIM provides requires a significant effort at operations and support to keep the resource responsive to community project needs. This includes the planning and management of physical facilities such as machine rooms and communications, system operations routine to backup and retrieve user files in a timely manner, and user support for communications, systems, and software advice. Of course, the upgrade of the KI-TENEX system to the 2060 required major planning and care to ensure continuous resource operation during the phase-over. Similarly, the relocation of our VAX 11/780 to Pine Hall and the outfitting of the KSL machine room at the Welch Road laboratory required much effort.

We use students for much of our operations and related systems programming work. Over the past 4 years, we have hired and trained a total of 15 undergraduate operations assistants.

We also spend significant time on new product review and evaluation such as Lisp workstations, terminals, communications equipment, network equipment, microprocessor systems, mainframe developments, and peripheral equipment. We also pay close attention to available video production and projection equipment, which has proved so useful in our dissemination efforts involving video tapes of our work.

2.1.4.5. Core AI Research

We have maintained a strong core AI research effort in the SUMEX-AIM resource aimed at developing information resources, basic AI research, and tools of general interest to the SUMEX-AIM community. It should be noted that the SUMEX resource grant from NIH provides much of the computing environment for this core AI work[1] but NIH supports only a small part of the manpower and other support for core AI. For example, NIH has provided partial funding for work on the AI Handbook, the AGE project, and part of the core ONCOCIN development for the dissemination of consultative AI systems. Substantial additional support for the personnel costs of our core AI research (roughly comparable to the NIH investment in computing resources) comes from DARPA, ONR, NSF, NASA, and several industrial basic research contracts to the Knowledge Systems Laboratory or KSL[2] (see the summary of core research funding on page 105).

Our core AI research work has long been the mainstay on which our extensive list of applications projects are based. This work has been focused on medical and biological problems for over a decade with considerable success, particularly in the area of expert systems which represent one important class of applications of AI to complex problems -- in medicine, science, engineering, and elsewhere. Numerous high-performance, expert systems have resulted from our work on expert systems in such diverse fields as analytical chemistry, medical diagnosis, cancer chemotherapy management, VLSI design, machine fault diagnosis, and molecular biology. Other projects have developed generalized software tools for representing and utilizing knowledge (e.g., EMYCIN [6, 68], UNITS [66], AGE [54], MRS [20], GLISP [57]) as well as comprehensive publications such as the three-volume *Handbook of Artificial Intelligence* [1] and books summarizing lessons learned in the DENDRAL [43] and MYCIN [6, 65] research projects.

But the current ideas fall short in many ways, necessitating extensive further basic research efforts. Our core research goals are to analyze the limitations of current techniques and to investigate the nature of methods for overcoming them. Long-term success of computer-based aids in medicine and biology depend on improving the programming methods available for representing and using domain knowledge.

The following summary reports progress on the basic or core research activities within the KSL. As indicated earlier, the development of the ONCOCIN system (under Professor Shortliffe) is an important part of our core research proposal for the renewal period. Progress on that work is reported separately in Section 6.1.3 on page 209, however, because its efforts have been supported as a collaborative and resource-related research project up until now. Together, this work explores a broad range of basic research ideas in many application settings, all of which contributes in the long term to improved knowledge based systems in biomedicine.

Recent Highlights of Research Progress

Research has progressed on several fundamental issues of AI. As in the past, our research methodology is experimental; we believe it is most fruitful at this stage of AI research to raise questions, examine issues, and test hypotheses in the context of specific problems such as management of patients with Hodgkins disease. Thus, within the KSL

---

[1]DARPA funds have also helped substantially in upgrading the KI-TENEX system to the 2060 and in the purchase of community Lisp workstations

[2]See Appendix A on page 285 for an overview of the KSL organization.

we build systems that implement our ideas for answering (or shedding some light on) fundamental questions; we experiment with those systems to determine the strengths and limits of the ideas; we redesign and test more; we attempt to generalize the ideas from the domain of implementation to other domains; and we publish details of the experiments. Many of these specific problem domains are medical or biological. In this way we believe the KSL has made substantial contributions to core research problems of interest not just to the AIM community but to AI in general.

In addition to the technical reports listed later, the following books and survey articles were published just during this year -- 11 books total have been published in the past 4 years as indicated in Appendix A. These are of central interest to AI researchers and of direct relevance to the mission of the SUMEX-AIM resource.

BOOKS:

1. Buchanan, B.G. and Shortliffe, E.H., eds. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Reading, MA: Addison-Wesley Publishing Company, 1984.

2. Clancey, W.J. and Shortliffe, E.H., eds. *Readings in Medical Artificial Intelligence: The First Decade.* Reading, MA: Addison-Wesley Publishing Company, 1984.

3. Cohen, Paul R. *Heuristic Reasoning about Uncertainty: An Artificial Intelligence Approach.* London and Marshfield, MA: Pitman Advanced Publishing Program, 1985.

SURVEY ARTICLES: HPP 84-15, 84-20, 84-23, 84-28, and 84-32.

In addition, work is progressing on a textbook for students beginning to study medical computing and artificial intelligence[1]. This multi-authored volume should be completed in draft form by the end of 1985 and a 1986 publication date is contemplated. Writing this new book will be facilitated by the SUMEX resource, much as the *Handbook of AI* was in the past. A multi-authored text of this type, particularly one for which the authors are spread at numerous different universities around the country, would be a nightmare to compile if it were not for the SUMEX resource. Many of the contributors to the book have been assigned SUMEX accounts for purposes of manuscript preparation. On-line manuscript work through the shared facility, coupled with messaging capabilities, will greatly enhance the efficiency and accuracy of the developing chapters and the editing process.

Progress is reported below under each of the major topics of our work. Citations are to KSL technical reports listed in the publications section.

1. *Knowledge representation*: How can the knowledge necessary for complex problem solving be represented for its most effective use in automatic inference processes? Often, the knowledge obtained from experts is heuristic knowledge, gained from many years of experience. How can this knowledge, with its inherent vagueness and uncertainty, be represented and applied?

   A working version of NEOMYCIN has been implemented which demonstrates the effectiveness of representing strategy knowledge explicitly. A detailed study of rule-based systems was published in book form. Specific representational issues in logic-based systems were addressed in the

[1] Shortliffe, E.H., Wiederhold, G.C.M., and Fagan, L.M.; *An Introduction to Medical Computer Science*, Reading, MA: Addison-Wesley (in preparation).