based network communications in both the Lisp and SystemTool (Lisp installation) environments. It also provided several UNIX compatibility improvements. As part of the new release, we beta-tested other revised software modules, such as Xerox ROOMS, and updated all of our Lispusers modules for re-release as well as added some new ones described further on. Since this release occurred around the time our user community moved onto the TCP/IP-only SUN 4, we incorporated the TCP networking ability into our full lisp system, instead of a separate loadup and later made this the case as well with the previous release, Lyric. We also introduced a PostScript driver into our full loadup at the same time in accord with our changing printer situation. Since Medley proved to be more stable than Lyric, we shortly dropped support for Lyric and continue to use Medley along with the InterLisp-only Koto environment for performance reasons.

The new Medley/Maiko system was introduced at AAAI (for which we provided a demonstration application) along with a spin off company, ENVOS, that Xerox launched to develop and market their AI workstation products. ENVOS, by being a smaller, more aggressive company, was intended to solve a number of problems that Xerox AI systems division had experienced marketing their products. Though there was much early excitement in the technical press over this new company, late last year they were laying off employees and by early in the second quarter of this year the company was out of business. Since we had made a decision early on not to continue to support use of this environment in our laboratory, these events had little more than passing interest to us as we were not actual customers of ENVOS, except for a Xerox University Grant program (UGP) connection very near the end. The commercial future of Xerox Lisp remains unresolved.

Although in the previous annual report we were anticipating going off the extended Xerox maintenance agreement provided by the UGP, this maintenance agreement was extended for one more, final year. The agreement, which covers the Xerox file, print and boot servers as well as 15 1108/9 Lisp processors, will end shortly, as of May 31st of this year. Based on our current reduced usage and funding constraints we will not be extending this contract further. All of the Xerox servers were upgraded to the new Services 11.0 over the course of the year. This lead to a problem where older releases of Xerox Lisp could no longer access the printer. After tracking down the problem, a fix was quickly provided by the vendor. Additionally, the Xerox boot service we had been running which is vital to the installation and maintenance of the equipment was not provided under the new release of services. We are still trying to resolve this issue with Xerox, before our maintenance agreement expires. In the mean time, we have been running our Lisp-based boot server which we implemented several years ago before Xerox made this service available on their servers. Unfortunately, this server is significantly slower than the Xerox Mesa-based one and our maintenance operations on the Xerox equipment have been much less efficient.

We continue to operate half of our 1186s (13 out of 24), nearly all of our 1109's (11 out of 15) and none of our 1108's (15). We obtained a donation of six 1108/9's that were used for parts to repair broken machines and upgrade others from 1108's to 1109's. We are currently supplying two 1186 machines to projects outside the KSL and have provided three 1109's to a Molecular Biology project in which several MIS students participate. We are using one 1186 as a network monitor, running the Xerox Mesa development environment, at our Welch Road facility. We are also still using our Xerox 1132, though it is scheduled to be removed from use shortly when we remove an obsolete 3 MB experimental Ethernet. In order to facilitate the video recording of demonstrations of projects that have been completed on the Xerox equipment, partially in anticipation that such equipment will not be available for live demonstrations in the future, we retuned an 1108/9 display to be compatible with the video tape scan rate.

Although the 1186 is a more recent product than the 1109 we still use 1109's as they are equally powerful, more reliable and were, until recently, under service contract. The 1186 has proved somewhat less reliable though we continue to use as many as are fully functional and tend to use them in either locations that are further removed from our main building as they are significantly easier to transport or for applications requiring larger screens. We removed from service 15 1108's that were not part of the UGP, replacing them with either unused UGP 1109's or 1186s. The 15 unused 1108's were then collected together, tested and repaired in anticipation of selling them (at less than 5% original cost) back to Xerox. Unfortunately, despite the effort in preparing the machines, this arrangement fell through and we are still in the, potentially fruitless, process of finding a buyer.

As part of our transition away from the Xerox Lisp environment, we developed a tool to allow graphics in various forms to be transferred to the Apple Macintosh with minimal loss of information and maximum flexibility. It was important to both provide a way to transfer drawings as well as provide a way for users writing new documents using the Macintosh to include illustrations derived from work done using the older system (e.g. the ONCOCIN technical specification documents). We initially experimented with, and put some programming effort into, a utility provided by another University which did conversions between Xerox bitmaps and MacPaint documents. This utility basically worked but was limited to MacPaint's relatively small image limit (576 by 720 pixels) and could only handle bitmaps, not higher level graphic representations.

To overcome this first obstacle, limited bitmap size, we wrote a new utility that could convert Xerox bitmaps into Macintosh PICT bitmap format, one operator in the PICT graphics representation. Using this, we were able to transfer bitmaps larger than the screens of any of the systems we currently use. Once the files were converted and transferred to the Macintosh, some minimal fix-up was then needed to set the proper file type before the file could be read into one of several commercial and shareware applications.

This utility was later incorporated as a small piece of a much more ambitious program that addressed the second problem, high level graphic information transfer.

To convert most types of graphic data from the Xerox Lisp environment to the Macintosh required the ability to do it in such a way that the data was still manipulatable as abstract graphic entities (lines, curves, text, bit images, etc.). The PICT representation included these graphic abstractions, was reasonable, and sufficient for our needs. The biggest challenges to the technical and visual success of such a conversion were the problems related to the manipulation and mathematics of moving between the Xerox and Macintosh font representations, so this problem was attacked first.

With the intention of building a general purpose graphics conversion program, we wrote a utility that allowed the Xerox generic graphics driver to read Macintosh font files and do font metric calculations. As a side benefit the utility also extended the number of display fonts available in the Xerox Lisp environment. Once completed, this program was made available separately to the national Xerox Lisp community as READAPPLEFONT, part of the Medley release Lispusers modules. Once the problem of font manipulation was in hand, work began, sometime later, on the full conversion program.

The next step involved implementing a full graphics driver for the PICT format based on our earlier graphics drivers (Impress, HPGL, CDI, etc.). This driver appeared to the user on the Xerox Lisp machine as printer file type (similar to PostScript, Interpress, etc.) but in fact operated at a higher level than any of our previous graphics drivers. This driver preserved grouping information so that dashed lines, polygons, etc. did not get broken into individual lines as usually happens when a document is converted into a laser printer master. The graphics driver took more time to debug, fine tune and polish than earlier ones as it incorporated much more detail (including color and 16-bit to 8-bit character set conversions) and the Macintosh proved relatively hostile and uninformative when it came to debugging PICT data files. The READAPPLEFONT module was also further refined to handle subtle issues regarding bold and italic font widths.

Once operational, the PICT conversion program was fine tuned by transferring existing graphics documents obtained from as wide a range of sources as possible, evaluating the results and modifying the behavior of the program accordingly. A large collection of Macintosh font files were extracted into Xerox Lisp readable files to facilitate this. Once converted to PICT representation, the Xerox graphics were able to be read and, most importantly, manipulated using program such as MacDraw I & II, Canvas, GrayView, Giffer, etc. This process also gave us another means to evaluate the various graphics drawing programs available for the Macintosh in our search for a reasonable one for laboratory-wide use. The completed program was used to convert numerous documents, bitmaps and other graphics entities and continues to be used on a regular basis.

Some student time to do conversions was provided for users who had documents that need to be transferred but who have distanced themselves from the Xerox Lisp environment. To further assist users, even though the program was developed under the Medley release of Xerox Lisp, it was patched and made available under the earlier Lyric release which was still in use at the time. The conversion program was released early in this year to the national Xerox Lisp community as the PICT Lispusers module.

A project was undertaken by a student programmer to implement a set of Common Lisp routines to index and interrogate a dictionary database. This project was done in Xerox Common Lisp and later tested under Allegro Common Lisp on the Macintosh. This package was then incorporated into a simple, experimental server that conformed to the TCP/IP Finger protocol. The host on which the server ran, a dedicated Xerox 1109, was aliased with the name Webster and users were then able to do a dictionary lookup from any of our hosts by doing: *finger word@webster*. This provided a simple network-wide dictionary facility without the overhead of storing the large dictionary database on each host that needed nor the effort of implementing new client software for each type of host. For debugging purposes, a Finger client implementation was done for the Xerox Lisp environment as well. The server continues to be used on an experimental basis.

Another project involved building a remote screen facility to allow one Xerox workstation to open a window onto another via the Ethernet to provide remote access and control of the screen, keyboard and mouse. Intended for evaluation purposes only and not for actual use on the Xerox workstations, the program was developed there using the superior networking and prototyping facilities. This program is described in more detail in the remote/distributed graphics portion of this report.

(6.6) Symbolics Lisp Machines

We have terminated all support of Symbolics workstations. As has been stated previously, in order for workstations to be competitive with time-shared mainframe computing resources, they must not only have a low purchase price, but must be cost-effective to maintain. This goal is normally achieved due to the economies of scale associated with having a large number of identical parts in an installation, as well as amortizing the cost of software development over many machines. We have come to reasonable agreements with all of the workstation vendors except for Symbolics. The high costs of service, the exceptionally high price of mail-in board repair, and the lack of a reasonable self-service alternative for the L-series workstations (3600, 3640, and 3670) has left us unable to justify continued support of these machines.

We are reluctant to get involved with one of Symbolics' newer products for three reasons. First, our experience with the L-series machines makes us wonder of Symbolics will drop support for current equipment as rapidly in the future. Second, we are leery of the occasional indications that Symbolics might not be around for much longer. Finally, there is little to recommend

Symbolics' products over their competitor's. However, Symbolics is still a player, and we will track their developments accordingly.

(6.7) HP 9836 Workstations

Three of the four HP 9836 workstations, which had been taken out of use, were provided to Dr. Craig Miller's group in Cardiovascular Surgery here at Stanford on indefinite loan. Dr. Miller's group already had similar equipment and was in need of the extra CPU cycles and keyboards our unused equipment could provide. We shut off the remaining HP 9836 after the arrival of the NeXT workstations as the HP/UX workstation was no longer needed for UNIX cycles. We retained the external Conrac color displays which were part of the HP equipment donation and retuned them to be compatible with our Macintosh workstations for use when a multiple display presentation or color is required.

(7) Remote Workstation Access, Virtual Graphics, and Windows

(7.1) Remote Access

The move towards a distributed workstation computing environment for AI research in the SUMEX-AIM community means that a number of technical obstacles must be overcome. For example, in order to allow users to work on workstations over networks from any location — at work, at home, or across the country — we must facilitate connecting a user display remotely with a workstation computing engine. The first step has been making reliable terminal access operational on all workstations, based uniformly on TCP-IP protocol services. This allows primitive (non-graphical) access between workstations. A more comprehensive access can be provided through remote virtual graphics connections.

(7.2) Virtual Graphics and Windows

In order to link the output of workstation displays across networks, it is necessary to capture and encode the many graphics operations involved so that they can be sent over a relatively low-speed network connection with the same interactive facility as if one had the display connected through the dedicated high-speed (30 Mhz) native vendor display/workstation connection. A mechanism for doing this is called a *remote graphics protocol.*

The *X window* system[1], developed under the MIT Project Athena, has become a widely touted remote graphics protocol standard. X is a very complete protocol that operates in a client/server fashion, where *server* in X terminology refers to the program running on the user's display and *client* refers to the program running on the remote host/workstation. Despite its wide publicity, X is not being adopted as the *core* window/graphics model by

---

1 Scheifler, R. W., and Gettys, J. "The X Window System." *ACM Trans. on Graphics*, , 1986.

many of the major workstation vendors. For example, Apple continues to use a proprietary protocol, NeXT has adopted Display PostScript as its imaging model, and SUN is still ambivalent between NeWS and X. Nevertheless, working X Windows implementations are available now for SUNs, MicroVAX's, the DEC MIPS workstation series, and soon will be available on Macintosh's.from a third party vendor. Common Lisp X client and server implementations also have now been released for TI Explorers (CLX) and include a supporting CLX library for the creation of windows, menus, scroll-bars and other graphical objects. Further, The Common Lisp User Environment (CLUE) is also available from TI and is a higher-level window system on top of the primitive CLX library which uses the Common Lisp Object System (CLOS). Since CLUE is a more general window-system/user-environment, it satisfies the long standing need for a portable window system for developing Common Lisp AI applications and will be applied here at SUMEX-AIM, for example in the porting of the ONCOCIN system from InterLisp to Common Lisp.

Because of the severe funding pressures this year and the need to devote almost all of our resources to the transition from the DEC 2060 to the SUN-4 computing environment, we have not been able to play a very active role in developing remote graphics access and related general system and user tools to our computing environment. Our early work on integrating the Stanford University V window system, and its virtual graphics terminal protocol based on structured display files, in the operating system of a Xerox 1186 and on writing an initial X client for the Explorer using the alpha release of the X.11 specification layed a good foundation. While diverted to our other tasks, workstation vendors have provided us with a more extensive and highly integrated set of tools. Our emphasis in the area of remote access/graphics has evolved from proving the concept of remote windows and remote tools to building real systems on top of remote access capabilities for routine use.

(7.3) Remote Graphics Applications

DISPLAYWATCH

We made a number of experiments with the *TIMBUKTU* remote machine manipulation program for the Macintosh which allows control via the network of the keyboard and mouse as well as remote viewing of the display. When used with a small screen Macintosh as the remote host over combinations of *AppleNet* and Ethernet segments, the program was somewhat sluggish. However, when tried between two machines with Ethernet boards, the response was quite good (as good as a smaller Macintosh when used directly). We also tested it between a staff member's home Macintosh and a Macintosh at Stanford via the *Shiva NetSerial* gateway. This worked, to the *NetSerial's* credit, but was far too slow when using a 2400 baud modem.

There were some problems with the software and a few annoyances that would make using it routinely a problem, particularly the unique serial numbered copies aspect — understandable, but it makes building servers on other types of machine much more difficult. Although it worked with our 19 inch and smaller displays, it did not work with our 24 inch displays despite the fact was supposed to handle any size display. To further experiment with the style of graphics protocol employed by this software, we built an equivalent of the program using the Xerox Lisp environment. This program was not compatible with the Macintosh one, but rather was optimized for the Xerox workstation as the program needed to be as fast and efficient as possible.

This system, *DISPLAYWATCH,* allowed a user to open a window on one Xerox workstation that would become the screen for another via the Ethernet and facilitate control of the remote keyboard and mouse using the local ones. Within the window, everything worked as if you were on the remote machine. The goal was to put together a working remote display system, by whatever means, to allow experimentation. The system took the approach that remoting everything on the Xerox workstation screen using a high level graphics protocol was impossible due to the fact that too many programs manipulated the screen bitmap directly, this was also true of the Macintosh. So, instead the program used a very low level approach that looked for bit (word) changes on the screen and sent update information across the network.

The system was layered on a reliable byte stream and worked using either the *SPP (XNS) or TCP/IP* protocols, though other types of byte streams could also be added (like *RS-232*). The system was nearly 100% functional, as compared to the local display, keyboard and mouse, and was reasonably stable. It included such details as interrupt character processing and control over remote and local mouse clicks. The only real problem was that when used between 1186 and 1109 class machines it was very slow. This was due to the CPU intensive server trying to find changes on the remote screen bitmap, not a communications problem. We further tested it using the faster Xerox 1132, in hope of potentially using this program as a solution to increasing the speed of the ONCOCIN clinic system without introducing a larger piece of hardware into the clinic. Unfortunately, although the improvement in response was quite encouraging, the program was not fast enough to solve the clinic problem.

This program was later demonstrated to several systems people from Xerox and ENVOS in hope of their taking on the task of further development and integrating it at a much lower level into their environment to provide the necessary speed. Unfortunately, though interested, they lacked the resources to take on such a project and development on the program has concluded. What the program was able to achieve, however, has helped influence our thinking regarding both the applications of remote graphics as well as the variety of implementations possible.

TALK

The TALK system (intra-machine user communication tool which employs both text and graphics) was substantially reorganized and had several improvements added. The previous release of the TALK program was contained in a single module that would detect what was contained in the environment (*TEdit, TCP,* etc.) and make services available accordingly. The new TALK was completely broken out into individual modules (three services, three network protocols and the main program) which could be loaded independently based on need. This fixed a number of problems and only introduced a much smaller set of others. Several bugs related to the new release of the Xerox Lisp environment, *Medley,* were also tracked down and solved (some by the vendor).

As part of convincing ourselves that the program was layered correctly, we added a *PUP* protocol module. We were able to get this up and running quite quickly without any modification to the TALK program itself. This protocol used a simple *PUP* packet exchange protocol to negotiate the connection and then switched to using a *BSP* byte stream. This was only meant to test the design of the TALK program and the additional protocol was not released to the Xerox Lispusers community. TALK had always determined which service (*TTY, TEdit or Sketch)* to use based on what the two communicating machines had in common. However, previously, it would map over the known network protocols (*XNS and TCP/IP*) and just use the first one that was able to resolve the destination name into an address, and if there was no server on the other side, it would then quit. It was improved to be able to try all the protocols (*XNS, TCP/IP and PUP*) that resolve the destination name in turn to see if it could locate a server with any of them.

Despite the reorganization, TALK remained identical at the network interface so the revised TALK was completely compatible with older versions. The user interface was kept nearly identical as well. The Xerox Lisp implementation work on TALK has been concluded and the revised version was provided to the national Xerox *Lispusers* community as part of the *Medley* release. There are no plans to continue work on this implementation.

Other Remote Tools

Last year we finished a tool (MONITOR) that allows a Xerox Lisp machine to examine the display of another workstation. This tool shows a shrunken version of the entire remote machine's display in one window and allows you to examine a smaller portion of the display in full size in another. We still plan to use this to remotely examine the display on the Oncology clinic machine (which runs the ONCOCIN expert system) when one of the physicians calls up with a problem. This tool is another built on top of the Courier server we described in previous years.

(8) Network Services

An 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.

(8.1) National and Wide-Area Networks

ARPANET Status

Since the early 1970's, the ARPANET has been the primary link between SUMEX and other university and AIM machine resources, including the large AI computer science community supported by DARPA. It provided key tools for collaboration and software sharing, including electronic mail transfer, remote terminal connections, and other file transfers.

Major changes have been underway over the past year in the allocation and management of ARPANET resources. DARPA has sought to limit the funding it allocates to network operations in order to focus more of its efforts on networking research and long-term developments. It does not want the mandate of organizing or running a broad national research Internet. In the spring of last year, DARPA began the reconfiguration of the ARPANET to eliminate all connections other than those directly related to DoD applications. An implication of these changes has been the elimination of almost all university ARPANET connections and their replacement by NSFNet links. Early in April of 1989, the connection to our DECSystem-20 was deactivated. Although ARPANET equipment is still on-site to continue the basic trunk-line connectivity of the network, it will soon be removed after ARPA contractors finish reconfiguring the network in the bay area to a smaller scale.

In the longer term, DARPA's role will be limited to DoD communications research and support of its contractor community. Thus, the ARPANET has once again become very restrictive in terms of access and use. DARPA expects to construct a Defense Research Internet (DRI) over the next several years (in cooperation with DCA and SDI), with the current ARPANET completely disappearing after that.

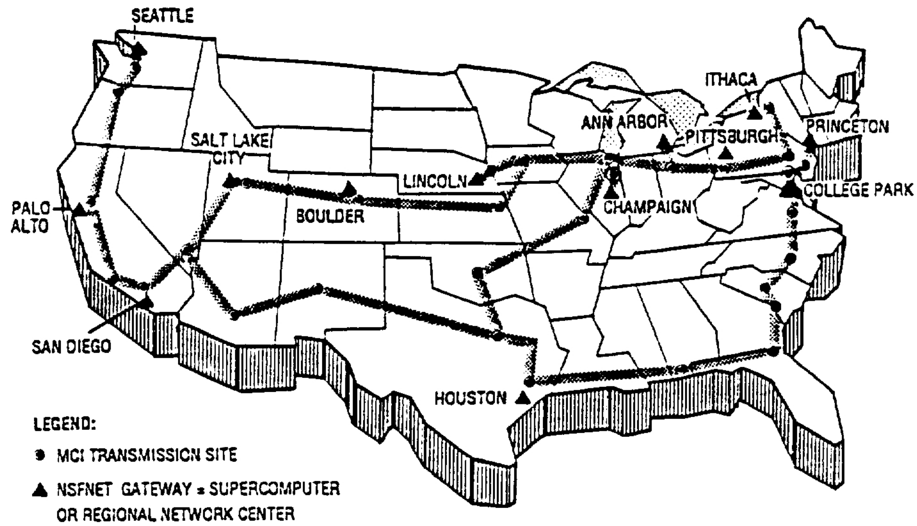

Figure 1. NSFNet Configuration as of January 1989 (from Marshal)

NSFNet Status

The NSFNet has been broadly announced as a national research network[1] and consists of a national backbone network coupled to a number of regional networks that link research institutions with each other and with the backbone (see Figure 1). The backbone operates with T1 (1.5 Mbit/sec) links between nodes. The current network is being managed by MCI, Inc. with MERIT Corp. (a nonprofit consortium of 8 Michigan universities) and IBM as research partners. The backbone links the 6 current NSF supercomputer sites and 7 regional academic computing networks.

The Bay Area Regional Research Network (BARRNet), to which the Stanford University Network (SUNet) is connected, now provides the primary link between SUMEX-AIM and the national Internet community. The upgrade of the backbone services to T1 links has improved network throughput and responsiveness significantly over the previously highly-congested 56 KBit/sec ARPANET services. The NSFNet/BARRNet services are currently funded by NSF and Stanford University so that NIH derives the benefits without any direct investment, as was the case with the earlier SUMEX-AIM connection to the ARPANET which was funded by DARPA as part of its support of KSL basic research activities.

1 Marshal, E. "NSF Opens High-Speed Computer Network." Science. 243: 22-23, 1989.

TELENET Service

SUMEX continues to be served by the commercial network, TELENET Inc. As reported previously the SUMEX connection to the TELENET is by means of a four-way gateway (developed by Develcon, Inc. — see Figure 7), which provides protocol translation between the X.25 system used by TELENET and our TCP/IP-based Ethernet system. Though somewhat experimental when set up, it has provided a reliable connection to TELENET. Though it is primarily for users coming into the SUMEX system, it also provides outbound connections. Thus users can be at their normal SUMEX terminal and access hosts on the TELENET system.

The Stanford Lane Medical Library catalog and circulation system (LOIS), which runs on a Tandem computer system, is connected to the Develcon gateway as is a leased line to the bibliographic search facilities of Dialog Inc. These links, under the direction of the Lane Library, provide SUMEX users with access both to the database of Lane Library materials and to the MEDLINE service provided by Dialog. The latter is a replacement for a similar service provided by BRS Inc. which was terminated in April of this year. As with the LOIS system described above, SUMEX users can also access the MEDLINE service much like they would other local services on the Ethernet.

In spite of the potential for greatly improved functionality and information resource access, the style of the interface between X.25-based nets and TCP/IP-based nets is not very smooth. Vastly different approaches to addressing, connection concepts, and terminal handling, make a general solution to the interface problem very difficult.

(8.2) Local Area Networks - LAN's

In the past, we have developed 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. A diagram of our local area network system is shown in Figure 8 and the following summarizes our LAN-related development work.

Workstation Networks

As noted in last year's report, most of our Apple Macintosh computers are connected with *PhoneNet* products. They are integrated with the rest of our equipment by connecting the PhoneNet networks to the campus Ethernet networks using *Kinetics FastPath* gateways, a commercial spin off resulting from the SUMEX work on the SEAGATE gateway. Kinetics licensed this software/hardware from Stanford University and used it as a basis for its product.

This year we added several Kinetics AppleNet-to-Ethernet gateways to support our Mac II's, LaserWriters, and MicroExplorers to link them with the Stanford Ethernet system and thereby to the NSFNet Internet. Kinetics introduced a new model gateway — the *FastPath-4* — which incorporated a faster processor, more memory, and new software. We delayed a portion of our Macintosh network installation last year so we could take advantage of this new model when it finally became available. Our benchmarks demonstrated that the new model improved throughput by factors of 2-10. By using Kinetics purchase credits, we may be able to replace our remaining model 1 and 2 FastPath gateways with model 4's this year.

AppleShare protocol file service for the Macintoshes is provided by two SUN file servers and a VAX file server running the CAP AppleShare file server software.

We also purchased a *Shiva NetSerial X-232* AppleTalk Dial-In server to experiment with dial-in network access. This device is discussed elsewhere in this report in the section titled "Distributed Information Resources and Access".

Ethernet Gateways

In our heterogeneous network environment, in order to provide workstation access to file servers, mail servers, and other computers within the Stanford local area network, it is necessary to able to route multiple networking protocols through the network gateways. As reported last year, the SUMEX gateways support PUP, Xerox NS, Symbolics/Texas-Instrument CHAOSNet, and the TCP/IP protocols. This support not only provides the routers necessary to move such packets among the subnetworks, but also other miscellaneous services such as time, name/address lookup, host statistics, bootstrap support, address resolution, and routing table broadcast and query information.

This year we began the transition away from the SUMEX developed gateways and TIPs to the Cisco Systems, Inc., gateway software and hardware. Cisco Systems' software products are derived from the SUMEX software as licensed through the Stanford University Office of Technology Licensing. The Cisco gateway servers also support the diverse protocols we use and are compatible with our environment. We were given Cisco processors by the Stanford IR-Networking group, as well as non-volatile RAM memory boards (NVRAMs). These systems have battery backup and contain configuration information essential for each gateway. Once the NVRAMs are initialized, the gateways can load their software from on-board PROM's and their configuration information from the NVRAMs after a power failure so that remote booting of these critical network nodes is no longer necessary. We believe this step will free our staff from more routine maintenance tasks so that they can concentrate their time on development.

Terminal Interface Processors

SUMEX-AIM has five TIPs, and in previous years we spent a significant amount of time maintaining and augmenting this software. Over the past year, as with our Ethernet gateways, we migrated the EtherTIP software to the Cisco Systems terminal processor software. This software is available under site license to Stanford University. It has many additional features such as Serial Line IP (SLIP) support, serial port dial out, and the NVRAM power failure protection feature mentioned above.

Currently, one of our TIPs has ten dial-in ports and is used extensively by the local SUMEX-AIM community for dial-in access to the SUMEX-AIM SUN-4, SUN-3, and VAX 11/750 servers from home during off hours. The four remaining TIPs are used to access various mainframes during work hours.

(9) Distributed Information Resources and Access

As mentioned in last year's report, there are many user needs for getting information from and about the computing environment, ranging from help with command syntax to sophisticated database queries. A distributed computing environment adds new complexities in making such information accessible and also new requirements for information about the distributed environment itself. We began to adapt the many workstation-specific information tools to include distributed environment information such as workstation and server availability, "Finger" information about user locations and system loads, network connectivity, and other information of interest to users in designing approaches to carrying our their research tasks. In addition, this year we wanted to develop general systems tools for monitoring and debugging distributed system performance to identify workstation and network problems. Finally, we still need to adapt and develop distributed system tools for remote database queries and organize the diverse sources of information of interest to AIM community members to facilitate remote workstation access to community, project, and personal information that has traditionally existed in ad hoc files on mainframe systems.

Only marginal progress has been made toward these goals this past year because NIH budget cuts forced us to place most of our efforts in the early part of this reporting period on shutting down the 2060 and moving the SUMEX-AIM users to the SUN-4. This move has been completed, but as a result, we have not had as much time to devote ourselves to the research needs required to support our distributed environment as we had hoped to have during this year's reporting period.

Still, the Macintosh HyperCard tool provides a very powerful environment in which to hierarchically organize and provide access to information in the form of text, graphics, pictures, and even sound. This year we began the development of a "KSL stack" for HyperCard that will eventually include descriptions of KSL and AIM personnel, SUMEX-AIM projects, SUMEX-AIM computing systems, KSL building maps, the KSL bibliography, etc. In response to a user request to share data in a HyperCard stack among several workstations, we investigated and developed a work-around that let them simultaneously access the stack on a SUN file server. When Apple releases HyperCard version 1.2.2 this work-around will no longer be necessary.

On another front, as we reported last year, in conjunction with the SUN file server we had mounted an experimental database system for remote information access using the commercial UNIFY database product. Our goal was to make access to the database information possible from a distributed workstation environment through network query transactions, as opposed to asking the user to log into the database system as a separate job and type in queries directly. This will facilitate remote information access from within *programs* including expert systems, where the information can be filtered, integrated with other information, and presented to the user. Such a system will provide multi-user, multi-database access capability; that is, several users will be able to have access to a single database at the same time, and a single user will be able to have access to several databases at the same time.

Since last year we have been investigating other database products that would more fully integrate our Macintoshes as well as Explorers and Micro-Explorers with databases resident on SUN file servers. There are other commercial databases besides UNIFY available, e. g., SYBASE, and front ends like CL/1 that will allow one to use slightly divergent client implementations of SQL in such a way that CL/1 will post the query to the appropriate database server, be it UNIFY or SYBASE. These clients still lack adequate TCP/IP support, and we in fact, have offered our TCP/IP stream package to these vendors to encourage its use in their products.

Also, as an experiment, we developed an English-language dictionary server on a Xerox 1108. Any host implementing the IP protocol *Finger* client can access the server without additional programming. (See the *Xerox Lisp Machines* section of this report for further details.)

As an experiment in remote network access for Macintosh workstations, we purchased a *Shiva NetSerial X-232* AppleTalk Dial-In server to give us dial-in network access. A researcher (or collaborator) with a Macintosh at another location (perhaps his home) can use the NetSerial software and a modem to connect his Macintosh (via the telephone connection and the NetSerial device) to one of the Phonenet AppleTalk networks at SUMEX. Then the researcher has access to SUMEX's file servers and printers as though his Macintosh were connected directly to the network. Owing to the low bandwidth of telephone lines, it has turned out that throughput discourages all but the most patient user. We experimented with TCP connections from a Macintosh at a private home to non-AppleTalk hosts at Stanford via the NetSerial. Although this experimental configuration worked surprisingly well, we plan to experiment with the SLIP protocol in the coming year. The SLIP protocol is potentially more efficient and is more widely implemented.

Finally, with a new version of the MacTCP-based *SU MacIP* TELNET program (which we are currently alpha testing) our users will be able to initiate dial-out terminal sessions from their Macintoshes to remote hosts via the networks and EtherTIP, which has 10 telephone lines and modems attached to it. The EtherTIP now has software to support this type of dial-out service. Using the current version of SU MacIP, users are already able to access the MEDLINE bibliography service provided by Dialog via our networks and the Develcon gateway.

(10) Distributed system operation and management

As mentioned in last year's report, the primary requirements in this area are user accounting (including authorization and billing), data backup, resource allocations (including disk space, console time, printing access, CPU time, etc.), and maintenance of community data bases about users and projects. Our accounting needs are a function of system reporting and cost recovery requirements. The distributed environment presents additional problems for tracking resource usage and will require developing protocols for recording various kinds of usage in central data base logs and programs for analyzing and extracting appropriate reports and billing information. We are still involved in analyzing the kinds of resource usage that can be reasonably accounted for in a distributed environment (e.g., printing, file storage, network usage, console time, processor usage, server access), and investigating what facilities vendors have provided for keeping such accounts. Data backup is, of course, closely related to the filing issue. We continue to use and improve network based file backup for many of our file servers.

III.A.2.5. Relevant Core Research Publications

The following is a list of new publications and reports that have come out of our core research and development efforts over the past year. In addition, we include references to earlier reports that are discussed in the "Progress Summary" text above.

KSL 87-22 Homer L. Chin and Gregory F. Cooper; **Stochastic Simulation of Casual Bayesian Models,** November 1988. To appear as a chapter: *Bayesian Belief Network Inference Using Simulation* , in the forthcoming book: *Uncertainty in Artificial Intelligence 3* , L. N. Kanal, T.S. Levitt, and J.F. Lemmer, eds., North-Holland. 20 pages.

KSL 87-50 (Journal Memo) Mark E. Frisse; **Searching for Information in a Hypertext Medical Handbook: The Washington University Dynamic Medical Handbook Project,** August 1987. 7 pages Has appeared in: Communications of the ACM, July 1988, Volume 31, Number 7, pp. 880-886.

KSL 87-70 Tu, S.W., Kahn, M. G., Musen, M.A., Ferguson, J.C., Shortliffe, E.H., and Fagan,, L.M; "**Episodic Monitoring of Time-Oriented Data for Heuristic Skeletal-Plan Refinement.**" August 1988, April 1989. 45 pages.

KSL 88-18 Peter D. Karp; **A Process-Oriented Model of Bacterial Gene Regulation,** June 1988. 14 pages

KSL 88-29 Andrew Gelman, Susan Altman, Matt Pallakoff, Ketan Doshi, Catherine Manago, Thomas C. Rindfleisch, and Bruce G. Buchanan; **FRM: An Intelligent Assistant for Financial Resource Management,** April 1988. 25 pages. *Proceedings of The Seventh National Conference on Artificial Intelligence*, pages 31-36, August 1988.

KSL 88-33 Bruce A. Delagi and Nakul P. Saraiya; **Elint in Lamina, Application of a Concurrent Object Language,** presented to: *Workshop on Object-Based Concurrent Programming-OOPSLA 88.,* July 1988, 13 pages.

KSL 88-35 Eric J. Horvitz; **Reasoning Under Varying and Uncertain Resource Constraints,** April 1988. *Proceedings of the Seventh National Conference on Artificial Intelligence, pp. 111-116, AAAI '88,* Minneapolis, Minnesota in August 1988., April 1988. 6 pages

KSL 88-37 (Working Paper) Liam Peyton, PhD; **A Transformational Approach to Software Redesign,** May 1988. 23 pages

KSL 88-39 Anthony Zygmont; **SOLACE: Systems Optimization Laboratory's Automated Computational Expertise,** May 1988. 55 pages.

KSL 88-40 Harold P. Lehmann; **Knowledge Acquisition for Probabilistic Expert Systems.** Published in *SCAMC Proceedings,* November 1988. 6 pages.

KSL 88-41 (Working Paper) Alan C. Noble and Everett C. Rogers; **AIRTRAC Path Association: Development of a Knowledge-Based System for a Multiprocessor,** June 1988. 101 pages

KSL 88-42 (Working Paper) Alan C. Noble; **ELMA Programmers Guide,** August 1988. 36 pages.

KSL 88-45 Susan M. Altman; **Representing and Editing Constraints: A Case Study in Financial Resource Management,** June 1988. 42 pages.

KSL 88-47 R. Martin Chavez and Gregory F. Cooper; **KNET: Integrating Hypertext and Normative Bayesian Modeling,** June 1988. 8 pages

KSL 88-50 Barbara Hayes-Roth, Micheal Hewett, Richard Washington, Rattikorn Hewett, Adam Seiver; **Distributing Intelligence Within an Individual,** October 1988. To appear in : *Distributed Artificial Intelligence*, Vol 2, L. Gasser and M.N. Huhns, (eds.), Morgan Kaufman, 1988. 23 pages

KSL 88-52 (Working Paper) John Sullivan; **RL3: An Approach to Incremental Rule Learning,** June 1988. 24 pages

KSL 88-57 Richard M. Keller; **Learning Approximate Concept Descriptions,** July 1988. 15 pages

KSL 88-58 David M. Combs, Samson W. Tu, Mark A. Musen, Lawrence M. Fagan; **From Expert Models to Expert Systems: Translation of an Intermediate Knowledge Representation,** August 1988, 13 pages

KSL 88-59 Mark A. Musen and Johan Van der Lei; **Of Brittleness and Bottlenecks Challenges in the Creation of Pattern-Recognition and Expert-System Models,** August 1988. 18 pages.

KSL 88-60 Thierry Barsalou, M.D.; **An Object-Based Architecture for Biomedical Expert Database Systems,** August 1988. Published in *SCAMC Proceedings,* November 1988. 8 pages.

KSL 88-61 Edward H. Shortliffe, M.D., Ph.D.; **Medical Knowledge and Decision Making,** September 1988. 14 pages.

KSL 88-62 Max Hailperin; **Load Balancing for Massively-Parallel Soft Real-Time Systems,** August 1988. 19 pages

KSL 88-63 Curtis P. Langlotz and Edward H. Shortliffe; **An Analysis of Categorical and Quantitative Methods for Planning under Uncertainty,** September 1988. Published in *SCAMC Proceedings,* November 1988. 6 pages.

KSL 88-64 Barbara Hayes-Roth; **Making Intelligent Systems Adaptive,** September 1988, 24 pages. Also appears in K. VanLehn (Ed.) *Architectures for Intelligence*. Lawrence Erlbaum, 1989.

KSL 88-66 Penny Nii, Nelleke Aiello, James Rice, **Experiments on Cage and Poligon: Measuring the Performance of Parallel Blackboard Systems,** to appear in ***Distributed Artificial Intelligence II.*** Pitman Publishing Ltd. & Morgan Kaufmann, 1989. February 1989, 69 pages.

KSL 88-69 James Rice,**The Elint Application on Poligon: The Architecture and Performance of a Concurrent Blackboard System,** December 1988. 11 pages.

KSL 88-70 Michael a. Shwe, Samson W. Tu, Lawrence M. Fagan; **Validating the Knowledge Base of a Therapy-Planning System,** published in:*Methods of Information in Medicine 28(1): 36-50*, 1989., April 1989. 16 pages.

KSL 88-71 James Rice; **"The Advanced Architectures Project"**. March 1989. 27 pages.

KSL 88-73 Mark A. Musen. **Generation of Visual Languages for Development of Knowledge-Based Systems.** Chapter in *Visual Languages,* Volume II (R. R. Korfhage, E. Jungert, and T. Ichikawa, eds.) , New York: Plenum, 1989. 26 pages.

KSL 88-74 Beverley Kane and Donald W. Rucker; **AI in Medicine.** *AI Expert*, pp. 48-55, November 1988. 9 pages.

KSL 88-75 Holly B. Jimison; **A Representation for Gaining Insight into Clinical Decision Models,** November 1988. 5 pages.

KSL 88-76 Leslie Lenert, Lewis Sheiner, Terrence Blaschke; **Improving Drug Dosing in Hospitalized Patients: Automated Modeling of Pharmacokinetcs for Individualization of Drug Dosing Regimens** Published in *SCAMC Proceedings,* November 1988. 6 pages

KSL 88-77 Homer L. Chin; **Case-Based Tutoring from a Medical Knowledge Base.** Published in *SCAMC Proceedings*, November 1988. 8 pages.

KSL 88-80 Nelleke Aiello; **Cage: The Performance of a Concurrent Blackboard Environment.** December 1988. 11 pages.

KSL 88-81 Gregory T. Byrd, Nakul P. Saraiya, and Bruce Delagi. **Multicast Communication in Multiprocessor Systems.** Submitted for publication to: *1989 International Conference on Parallel Processing,* January 1989. 19 pages.

KSL 88-82 Edward H. Shortliffe; **Testing Reality: The Introduction of Decision- Support Technologies for Physicians,** published as an editorial in *Methods of Information In Medicine,* 28: 1-5,1989. 6 pages.

KSL 88-83 Edward H. Shortliffe, Lawrence M. Fagan; **"Research Training in Medical Informatics: The Stanford Experience."** *Proceedings of the International Symposium on Medical Informatics and Education,* Victoria, B.C., May 1989. March 1989. 8 pages.

KSL 88-84 I. A. Beinlich, H.J. Suermondt, R.M. Chavez and G.F. Cooper. **The ALARM Monitoring System: A Case Study with Two Probabilistic Inference Techniques for Belief Networks.** Submitted to *AI in Medicine,* London 1989.

KSL 88-85 Nakul P. Saraiya. **A Shared Memory Lisp Package for CARE** January 1989. 7 pages.

KSL 88-86 Perry L. Miller and Glenn D. Rennels; **Prose Generation from Expert Systems. An Applied Computational Approach,** November 1988. *AI Magazine,* pp. 37-44, Fall 1988. 9 pages.

KSL 89-01 Eric J. Horvitz, David E. Heckerman and Gregory Cooper; **Reflection and Action Under Scarce Resources: Theoretical Principles and Empirical Study.** To appear in: *IJCAI-89*, May 1989. 17 pages.

KSL 89-02 Rich A. Acuff; **Performance of Two Common Lisp Programs on Several Systems.** January 1989. 30 pages.

KSL 89-03 Thierry Barsalou, R. Martin Chavez, Gio Wiederhold. **Hypertext Interfaces for Decision-Support Systems: A Case Study.** Submitted to: *Medinfo 89.* 6 pages.

KSL 89-04 Geoffrey Rutledge, George Thomson, Ingo Beinlich, Brad Farr, Michael Kahn, Lewis Sheiner, and Lawrence Fagan.; **VentPlan: An Architecture for Combining Qualitative and Quantitative Computation,** to appear in *IJCAI-89*, January 1989. 9 pages.

KSL 89-05 Barbara Hayes-Roth, Rich Washington, Rattikorn Hewett, Micheal Hewett, and Adam Seiver; **Intelligent Real-Time Monitoring and Control,** January 1989 15 pages. And in *Proceedings of the International Joint Conference on Artificial Intelligence*, IJCAI-89, Detroit, MI., 1989.

KSL 89-06 Richard Washington, Barbara Hayes-Roth; **Data Management in Real-Time AI Systems.** March 1989. 11 pages. And in *Proceedings of the International Joint Conference on Artificial Intelligence*, IJCAI-89, Detroit, MI, 1989.

KSL 89-07 Mark A. Musen; **Languages for Knowledge Acquisition: Building and Extending Models,** January 1989. *Proceedings of AAAI Spring Symposium on Knowledge System Development Tools and Languages,* Stanford, CA March 1989. 7 pages.

KSL 89-08 James Rice and Nelleke Aiello; "**See How They Run.The Architecture and Performance of Two Concurrent Blackboard Systems.**" To appear in: **Blackboard Architectures and Applications: Current Trends.** March 1989. 22 pages.

KSL 89-09 Mark A. Musen; **Widening the Knowledge-Acquisition Bottleneck: Automated Tools for Building and Extending Clinical Methods.** *AAMSI Proceedings.* February 1989. 7 pages.

KSL 89-10 Peter D. Karp, David C. Wilkins; **An Analysis of the Distinction Between Deep and Shallow Expert Systems.** To appear in: *International Journal of Expert Systems, 1989.* February 1989. 36 pages.

KSL 89-13 Harold P. Lehmann, M.D; **Review of the Uncertainty in AI Workshops.** April 1989. 24 pages.

KSL 89-14 R. Martin Chavez; **Hypermedia and Randomized Algorithms for Medical Expert Systems.** Submitted to *SCAMC 89.* March 1989. 19 pages.

KSL 89-15 Gregory T. Byrd and Bruce A. Delagi; **Support for Fine-Grained Message Passing in Shared Memory Multiprocessors**. To appear in: *Proceedings of the 5th Annual Computer Science Symposium, University of South Carolina,* April 7-8, 1989. 21 pages.

KSL 89-16 Nakul P. Saraiya, Bruce A. Delagi and Sayuri Nishimura, **Design and Performance Evaluation of a Parallel Report Integration System** Submitted for publication. April 1989. 23 pages.

KSL 89-17 H.J. Suermondt and G.F. Cooper; **Probabilistic Inference in Multiply Connected Belief Networks Using Loop Cutsets.** Submitted to: *International Journal of Approximate Reasoning. March 1989.* 33 pages.

KSL 89-18 Thierry Barsalou and Gio Wiederhold; **A Cooperative Hypertext Interface to Relational Databases.** Submitted to *SCAMC '89.* March 1989. 22 pages

KSL 89-19 D.W. Rucker and E.H. Shortliffe; **A Methodology for Implementing Clinical Algorithms Using Expert System and Database Tools.** Submitted to *SCAMC '89.* March 1989.

KSL 89-20 Brad R. Farr; **Decision-Theoretic Evaluation of Therapy Plans.** Submitted to *SCAMC '89.* March 1989. 10 pages.

KSL 89-21 John Reed; **Building Decision Models that Modify Decision Systems.** Submitted to *SCAMC '89.* March 1989. 18 pages.

KSL 89-23 H.L. Suermondt and M.D. Amylon, M.D.; **Probabilistic Prediction of the Outcome of Bone-Marrow Transplantation.** Submitted to *SCAMC '89.* March 1989. 9 pages.

KSL 89-24 E.J. Horvitz, D.E. Heckerman, K.C. Ng, B.N. Nathwani; **Heuristic Abstraction in the Decision-Theoretic Pathfinder System.** Submitted to *SCAMC '89.* March 1989. 25 pages.

KSL 89-25 D.E. Heckerman, E.J. Horvitz, B.N. Nathwani; **Toward Effective Normative Decision Systems: Update on the Pathfinder Project.** Submitted to *SCAMC '89.* March 1989. 26 pages.

KSL 89-26 Mark A. Musen; **Automated Support for Building and Extending Expert Models.** To appear in: a special issue of *Machine Learning.* May 1989. 29 pages.

KSL 89-28 Mark A. Musen and Johan van der Lei; **Knowledge Engineering for Clinical Consultation Programs: Modeling the Application Area.** Published in *Methods of Information in Medicine, 28-28-35,* 1989. March 1989 9 pages.

KSL 89-31 R. Martin Chavez and Gregory F. Cooper; **An Empirical Evaluation of a Randomized Algorithm for Probabilistic Inference.** Published in: *Fifth Workshop on Uncertainty in Artificial Intelligence,* April 1989, 13 pages.

KSL 89-33 Bruce G. Buchanan and Edward H. Shortliffe; **Advances in Expert Systems (A White Paper and Commentary).** Presented to *DARPA, February, 1989.* April 1989. 21 pages.

KSL 89-34 Michael G. Kahn, Lawrence M. Fagan, and Lewis B. Sheiner; **Model-Based Interpretation of Time-Varying Medical Data.** April 1989. 25 pages

KSL 89-41 Harold P. Lehmann; **A Decision-Analytic Model for Using Scientific Data,** submitted to *AAAI Workshop in Uncertainty in AI,* May 1989. 19 pages.

KSL 89-42 E.J. Horvitz, H.J. Suermondt, G.F. Cooper; **Bounded Conditioning: Flexible Inference for Decisions Under Scarce Resources.** May 1989. 21 pages.

Other Publications Not Yet in the KSL Report Series

1) Hayes-Roth, B., Hewett, M., Washington, R., Hewett, R., and Seiver, A. **Distributing intelligence within a single individual**. In L. Gasser and M.N. Huhns (Eds.) *Distributed Artificial Intelligence* Volume 2. Morgan Kaufmann, 1989.

2) Hewett, R., and Hayes-Roth, R. **Representing and reasoning about physical systems using generic models**. In J. Sowa (Ed.) *Formal Aspects of Semantic Networks*. Morgan Kaufmann, 1989.

3) Hayes-Roth, B. **Dynamic control planning in adaptive intelligent systems**. *Proceedings of the DARPA Knowledge-Based Planning Workshop*, 1989.

III.A.2.6. Resource Equipment

The SUMEX-AIM resource is a complex, integrated facility comprised of mainframes, servers, workstations, and networks illustrated in Figures 2 - 8. A key role of the SUMEX-AIM resource is to continue to evaluate workstations in order to keep up with the rapidly changing technology. This evaluation includes new hardware and software, 1) to provide superior development and execution platforms for AI research, 2) to experiment with systems practical for the dissemination of AI systems and their integration with other biomedical systems, and 3) to support the ancillary "office environment" (presently carried out on the SUN-4 and Macintosh's, following the phase-out of the DEC 2060). Thus far no single workstation has materialized that provides all the services we would like to see in support of these missions. This means that for the foreseeable future, we will utilize a multiplicity of machines and software to address the needs of AIM projects.

Systems based on the Motorola 68030 chip (e.g., Apple Macintosh II or NeXT workstations), the Intel 80286 and 80387 chips (e.g., IBM PS/1-4 machines), and other reduced instruction set computer (RISC) chips (e.g., the SUN SPARC or MIPS R-2000 chips), have Lisp benchmark data rivaling the performance of existing, specially microcoded Lisp machines (see Appendix B). It is still early to predict how this "race" will ultimately turn out and software environments play an equally important role with raw hardware speed in the decision. For now, the Lisp software environments on the "stock" machines are not nearly so extensively developed as on Lisp machines and conversely, the routine computing environments of Lisp machines (text processing, mail, spreadsheets, etc.) lag the tools available on stock UNIX machines.

We had been seeking an integration of both the Lisp machine and stock machine worlds. As discussed extensively in the progress section on Core Systems Research, these two capabilities came together as never before with the Macintosh II and microExplorer coprocessor systems.

(1) Purchases This Past Year

The relatively small amount of SUMEX-AIM money for new equipment purchases has been concentrated on upgrades to facilitate the move to the SUN-4 environment, experimental workstations, and server equipment needed for distributed system development. These purchases are paced carefully with the developments of higher performing, more compact, and lower cost systems. The NIH-funded purchases this past year are summarized below. Note that the very large purchase of Mac II workstations, TI microExplorer coprocessors, and the SUN-4 network server was funded almost entirely from *non-NIH* funding sources, even though a significant part of this equipment benefits the AIM community as a whole. Again this year, numerous purchases were made by Stanford research projects to support their work from non-NIH sources.

(#)	Device Purchased	Cost	Comments
(3)	Mac II CPU's, with 24" monitors, 20 MB disks, keyboards, and AppleNet connections.	$12,400	These machines were to finish outfitting staff desks with Mac workstations.
(2)	Mac II Ethernet boards	$900	Direct Ethernet connection boards for systems development work.
(1)	ImageWriter LQ printer	$1,000	Experimental, inexpensive, color impact printer
(3)	LaserWriter II NTX printers	$11,500	Printer upgrades to replace unreliable IMAGEN 12/300 printers that could not be made to process PostScript.
(36)	2 MB memory expansion kits - Macs	$21,200	Mac II memory upgrades to allow running MultiFinder and large programs (e.g., HyperCard).
(4)	Trailblazer stand-alone modems	$4,000	Experimental high-speed modems for remote Ethernet connections.
(4)	Multibus Ethernet Controllers	$5,200	Upgrades for key Ethernet gateway to prevent lost packets during heavy traffic.
(1)	SUN disk controller	$3,300	SUN-4 disk controller upgrade to increase performance.
(1)	140 Meg hard drive	$1,300	Mac II external disk to facilitate systems support and maintenance.
(2)	NeXT machines w/ 330 meg drives	$18,000	Experimental workstations to evaluate the hardware design and Interface Builder software tools.
(1)	Helical scan back-up subsystem	$3,400	Experimental high-density tape backup system for large file server support
(5)	Pagers for key systems personnel	$2,000	For improved communications with staff to handle system emergencies.

(1)	Cisco processor- 10 MHz	$2,000	Upgrade to EtherTIP system so we can run standard commercial TIP software.
(1)	Ricoh fax machine	$400	1/4 share of a jointly used FAX machine to improve communications with AIM community and other contacts
	Total Equipment Cost	$86,600	

(2) Current Subsystem Configurations

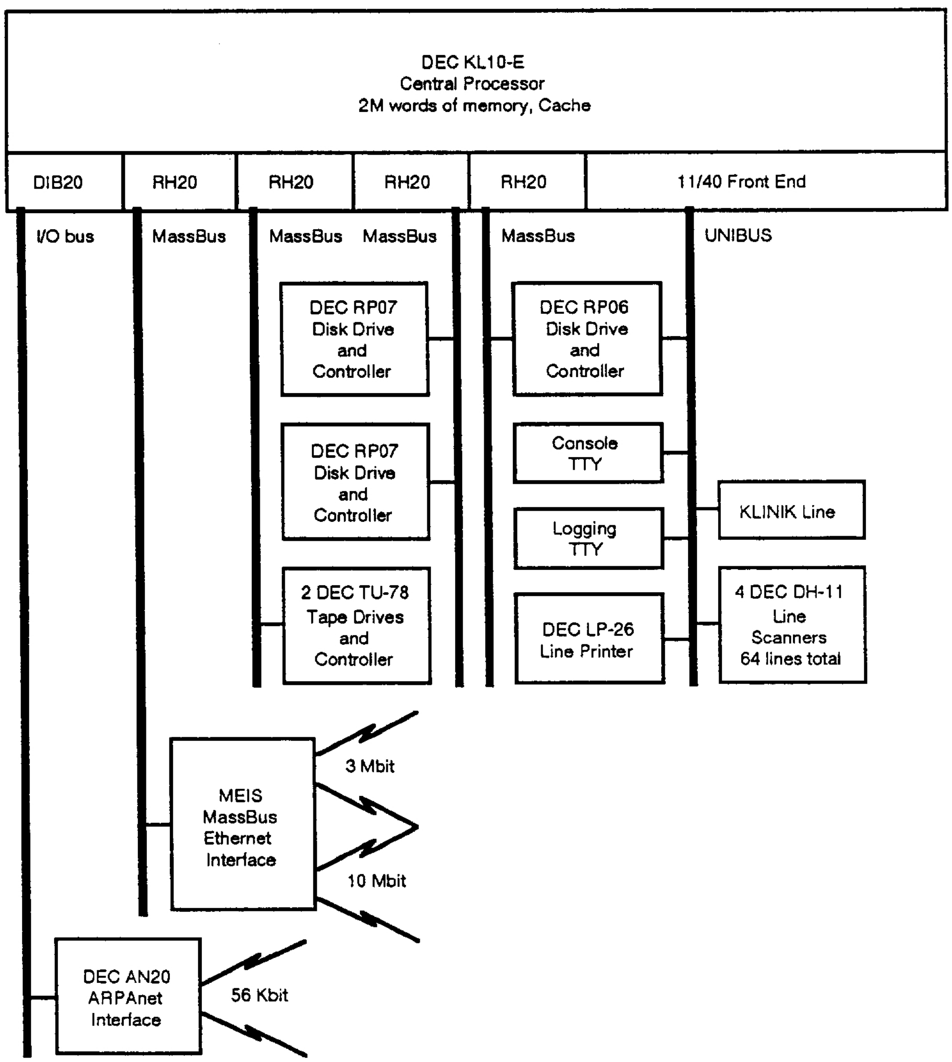

Figure 2. SUMEX-AIM DEC 2060 Configuration