

Table of Contents

Section	Page
List of Figures	iv
I. RESOURCE IDENTIFICATION PAGE	1
II. RESOURCE OPERATIONS	2
II.A PROGRESS	2
II.A.1 RESOURCE SUMMARY AND GOALS	2
II.A.2 TECHNICAL PROGRESS	4
II.A.2.a FACILITY HARDWARE	4
II.A.2.b TENEX SYSTEM SOFTWARE	12
II.A.2.c NETWORK COMMUNICATION FACILITIES	15
II.A.2.d SYSTEM RELIABILITY AND BACKUP	23
II.A.2.e PROGRAMMING LANGUAGES	25
II.A.2.f MAINSAIL OVERVIEW	27
II.A.2.g OPERATIONS AND USER SOFTWARE	30
II.A.2.h USER ASSISTANCE AND CONSULTING	36
II.A.2.i INTRA-COMMUNITY COMMUNICATION	37
II.A.2.j DOCUMENTATION AND EDUCATION	40
II.A.2.k SOFTWARE COMPATIBILITY AND SHARING	40
II.A.3 RESOURCE MANAGEMENT	44
II.A.3.a MANAGEMENT COMMITTEES	44
II.A.3.b NEW PROJECT RECRUITING	45
II.A.3.c STANFORD COMMUNITY BUILDING	46
II.A.3.d RESOURCE ALLOCATION POLICIES	46
II.A.3.e AIM WORKSHOP SUPPORT	47

II.A.4	FUTURE PLANS	48
II.B	SUMMARY OF RESOURCE USAGE	51
II.B.1	RELATIVE SYSTEM LOADING BY COMMUNITY	51
II.B.2	INDIVIDUAL PROJECT AND COMMUNITY USAGE	54
II.B.3	NETWORK USAGE STATISTICS	57
II.C	RESOURCE EQUIPMENT SUMMARY	59
II.D	PUBLICATIONS	65
III.	RESOURCE FINANCES	66
III.A	REFERENCE TO BUDGETARY DETAILS	66
III.B	RESOURCE FUNDING	67
IV.	RESOURCE PROJECT DESCRIPTIONS	68
IV.A	FORMALLY APPROVED PROJECTS	69
IV.A.1	STANFORD USERS	69
IV.A.1.a	DENDRAL PROJECT	69
IV.A.1.b	MYCIN PROJECT	89
IV.A.1.c	PROTEIN STRUCTURE MODELING PROJECT	100
IV.A.2	NATIONAL USERS	105
IV.A.2.a	CHEMICAL SYNTHESIS PROJECT	105
IV.A.2.b	INTERNIST (DIALOG) PROJECT	110
IV.A.2.c	HIGHER MENTAL FUNCTIONS MODELING	112
IV.A.2.d	LANGUAGE ACQUISITION MODELING PROJECT	118
IV.A.2.e	MEDICAL INFORMATION SYSTEMS LABORATORY	121
IV.A.2.f	RUTGERS COMPUTERS IN BIOMEDICINE	126
IV.B	INFORMAL PROJECTS	136
IV.B.1	STANFORD PILOT PROJECTS	136
IV.B.1.a	AI IN MOLECULAR GENETICS - MOLGEN	136
IV.B.1.b	BAYLOR-METHODIST CEREBROVASCULAR PROJECT	138
IV.B.1.c	AUTOMATIC LV MODELING	143

IV.B.1.d	INFORMATION PROCESSING PSYCHOLOGY PROJECT	147
IV.B.1.e	AIM RESEARCH - UNIVERSITY OF ROCHESTER	150
IV.B.1.f	QUANTUM CHEMICAL INVESTIGATIONS	151
IV.B.2	NATIONAL PILOT PROJECTS	154
IV.B.2.a	NATURAL LANGUAGE UNDERSTANDING	154
IV.B.2.b	KRL PROJECT	158
IV.B.2.c	COMPUTERIZED PATIENT MONITORING	159
IV.B.2.d	AI IN PSYCHOPHARMACOLOGY	160

Appendix A

OVERVIEW OF ARTIFICIAL INTELLIGENCE RESEARCH	166
--	-----

Appendix B

AI HANDBOOK OUTLINE	180
-------------------------------	-----

Appendix C

HEURISTIC PROGRAMMING PROJECT WORKSHOP	194
--	-----

Appendix D

TYMNET RESPONSE TIME DATA	195
-------------------------------------	-----

Appendix E

MAINSAIL DESIGN SUMMARY	208
-----------------------------------	-----

Appendix F

SUBSYSTEMS AND DOCUMENTATION DIRECTORIES	218
--	-----

Appendix G

SUMEX-AIM SUMMARY FOR ARPANET RESOURCES HANDBOOK	230
--	-----

Appendix H

AIM MANAGEMENT COMMITTEE MEMBERSHIP 252

Appendix I

USER INFORMATION - GENERAL BROCHURE 255

Appendix J

GUIDELINES FOR PROSPECTIVE USERS 264

List of Figures

1.	SUMEX-AIM Computer Configuration	9
2.	Elapsed Time/CPU Minute versus Load Average	10
3.	I/O Wait and Core Management Overhead versus Load Average	11
4.	TYMNET Network Map	19
5.	TYMNET Response Delay Statistics	20
6.	ARPANET Geographical Network Map	21
7.	ARPANET Logical Network Map	22
8.	CPU Usage by Community	52
9.	File Space Usage by Community	53
10.	TYMNET and ARPANET Usage Data	58

NATIONAL INSTITUTES OF HEALTH
 DIVISION OF RESEARCH RESOURCES
 BIOTECHNOLOGY RESOURCES PROGRAM

SECTION I - RESOURCE IDENTIFICATION

Report Period: From August 1, 1975 to July 31, 1976

Grant Number:

RR-00785-03

Report Prepared:

May, 1976

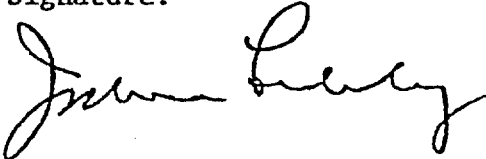
Name of Resource: Stanford University Medical Experimental Computer (SUMEX)	Resource Address: Stanford University Stanford, California 94305	Resource Telephone Number:
Principal Investigator: Joshua Lederberg, Ph.D.	Title: Chairman and Professor Department of Genetics	Academic Department: School of Medicine Department of Genetics
Grantee Institution: Stanford University	Type of Institution: Private University	Investigator's Telephone No.: (415) 497-5801

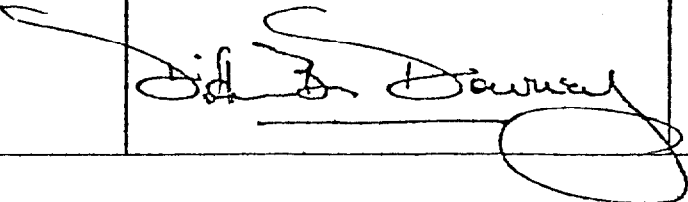
Name of Institution's Biotechnology Resource Advisory Committee:

SUMEX-AIM Executive Committee

Membership of Biotechnology Resource Advisory Committee:

<u>Name</u>	<u>Title</u>	<u>Department</u>	<u>Institution</u>
Saul Amarel, Ph.D.	Chairman and Professor	Computer Science	Rutgers University
Donald Lindberg, M.D.	Professor Director	Pathology Information Science Group	University of Missouri School of Medicine

Principal Investigator: Joshua Lederberg, Ph.D. Chairman and Professor	Signature: 	Date: May 25, 1976
--	--	---------------------------

Stanford University Official: D'Ann B. Downey Sponsored Projects Officer	Signature: 	Date: May 27, 1976
--	--	---------------------------

II RESOURCE OPERATIONS

II.A PROGRESS

II.A.1 RESOURCE SUMMARY AND GOALS

The SUMEX (Stanford University Medical EXperimental computer) project is a national computer resource funded by the Biotechnology Resources Program of the National Institutes of Health (NIH-BRP). It encompasses a dual mission: 1) the promotion of applications of artificial intelligence (AI) computer science research to biological and medical problems and 2) the demonstration of computer resource sharing within a national community of health research projects. The SUMEX resource resides administratively within the Genetics Department of the Stanford University Medical School and serves as a nucleus for a growing community of projects, both at Stanford and nationally. SUMEX provides computing facilities specifically tuned to the needs of AI research and communication tools to facilitate inter- and intra-group contacts and the demonstration of research products to medical users. The project also develops tools for and encourages community relationships among collaborating projects and medical researchers.

User projects are separately funded and autonomous in their management. They are selected for access to SUMEX on the basis of their scientific and medical merits as well as their commitment to the community goals of SUMEX (see Section II.A.3 on page 44). Currently active projects span a broad range of application areas such as clinical diagnostic consultation, molecular biochemistry, belief systems modeling, mental function modeling, and instrument data interpretation (see Section IV on page 68).

Artificial Intelligence is a branch of computer science which attempts to discern the underlying principles involved in the acquisition and utilization of knowledge in reasoning, deduction, and problem-solving activities(1). Each authorized project in the SUMEX community is concerned in some way with the application of these principles to medical problems. The tangible objective of this approach is the development of computer programs which, using formal and informal knowledge bases together with mechanized hypothesis formation and problem solving procedures, will be more general and effective consultative tools for the clinician and medical scientist. The exhaustive search potential of computerized hypothesis formation and knowledge base utilization,

(1) Two recent reviews give some perspective on the current state of AI: see Nilsson, N.J., "ARTIFICIAL INTELLIGENCE", Information Processing 74, North-Holland Pub. Co. and a summary by Buchanan, B. G. and Feigenbaum, E. A., attached as Appendix A, page 166. An additional overview of research areas in AI is provided by the outline for an "Artificial Intelligence Handbook" being prepared under Professor Feigenbaum by computer science students at Stanford (see Appendix B on page 180).

constrained where appropriate by heuristic rules or interactions with the user, has already begun to produce promising results in areas such as chemical structure elucidation, diagnostic consultation, and mental function modeling. Needless to say, much is yet to be learned in the process of fashioning a coherent scientific discipline out of the assemblage of personal intuitions, mathematical procedures, and emerging theoretical structure of the "analysis of analysis" and of problem solving. State-of-the-art programs are far more narrowly specialized and inflexible than the corresponding aspects of human intelligence they emulate; however, in special domains they may be of comparable or greater power, e.g., in the solution of formal problems in organic chemistry or in the integral calculus.

Our community building role is based upon the current state of computer communications technology. While far from perfected, these new capabilities offer highly desirable latitude for collaborative linkages, both within a given research project and among them. Several of the active projects on SUMEX are based upon the collaboration of computer and medical scientists at geographically separate institutions; separate both from each other and from the computer resource. Another major goal of the network experiment is to enable diverse projects to interact more directly and to facilitate selective demonstrations of available programs to physicians and medical students. Even in their current developing state, such communication facilities allow access to the rather specialized SUMEX computing environment and programs from a great many areas of the United States (even to a limited extent from Europe) for potential new research projects and for research product dissemination and demonstration.

This past year has seen much activity and growth in the SUMEX-AIM resource and community. Two new formal projects (one maturing from an earlier pilot effort) have been authorized to join the AIM community as well as several new pilot efforts. These new projects together with the increasing load from the earlier projects have raised the loading of SUMEX-AIM beyond productive limits, particularly during prime time. To accommodate this load, we received authority from the Executive Committee to augment the processing capacity of the system - implementation of this addition is in progress. Efforts have continued to build inter- and intra-group interactions through system communication facilities, workshops, a local "mini-conference" on AI techniques to pull together the Stanford community of projects, and a seminar project initiated by Professor E. A. Feigenbaum of Stanford to assemble from the community a handbook of AI concepts, methods, and state-of-the-art. There have also been continuing, substantial efforts by the community to introduce non-affiliated research people to a number of the programs which are far enough along in their development. The system staff has worked hard to maximize the computing resources and to enhance the repertoire of software available to the community and has investigated a variety of alternatives related to the import and export of operational programs. And finally, the management committees which help direct the allocation and development of the resource have been active in recruiting and evaluating new projects, planning future AIM workshops, and guiding the dissemination of resource objectives and opportunities to other medical institutions.

II.A.2 TECHNICAL PROGRESS

II.A.2.a FACILITY HARDWARE

Memory Swapping Drum System:

The hardware system has stabilized nicely over the past year, especially after the correction of a design flaw in the DEC-supplied drum controller(2). We had been having an abnormally high number of drum errors, mostly recoverable. The number exceeded the manufacturers' specifications and could not be explained by memory overruns or other contention problems. After much detective work (and vendor interactions), we discovered that a delay in the DEC drum controller between "drive select" and "sector ready" signals was too short to allow the read and write heads to settle down. After putting in the appropriate delay circuitry (in September 1975), the system has run to date without a single error (recoverable or permanent) or failure in the swapping system!

Computational Capacity:

A major event over the past year relating to system hardware was the decision to upgrade the central processor capacity. An updated diagram of the hardware configuration is shown in Figure 1 on page 9. As discussed in the last report, the high loading of SUMEX-AIM during prime time has restricted work. From a subjective viewpoint, the system became unworkably sluggish above a load average(3) of 4 or 5. We have made a number of administrative efforts to push as much work as possible into non-prime time. These have included excellent cooperation from user projects in encouraging programmers to work during night hours, doing operational functions (such as file system dumps) during the evening, and providing an effective batch system for running jobs in background mode and during non-prime hours. These steps have helped make better use of the non-prime hours but have not substantially relieved the midday congestion; the decreases achieved were offset by new development work and increased community use of operational AI programs (ONET, DENDRAL, MYCIN, and PARRY in particular). The principal period during which medical collaborators can, in practice, work with the programs remains the prime hours and our goal is to provide a computational capacity which allows reasonably interactive access to the programs at these times.

(2) We follow a long tradition in calling our fast, fixed-head disk a "drum" to distinguish it from the file system disks

(3) The "load average" signifies the number of jobs waiting in queue to be processed at a given instant: it measures the number of people awaiting service at that moment, so that responsiveness will be (approximately) inversely related to the load average. Two, three, or even four times as many users may be connected to the system at such times; but users typically take time out to ponder what the computer has reported, or the jobs may be preoccupied with input or output rather than the CPU.

We made a series of measurements to identify system bottlenecks(4) and observed a number of simultaneously limiting resources. A plot of the elapsed time required to accumulate 1 CPU minute is shown in Figure 2 on page 10 as a function of load average. A plot of the corresponding system I/O wait and core management overhead time is shown in Figure 3 on page 11. Data are plotted for a variety of jobs ranging from a one page CPU-bound job to a large, page-fault intensive jobs and include several FORTRAN and INTERLISP jobs. The data were not collected on a dry machine, they were run at night when the user load was low but not zero. Thus, some dispersion exists in the data. The key features of the data (and other internal system parameters) are that for small jobs the elapsed time-to-completion increases linearly with load average. That is at load average N , the CPU is split into roughly $1/N$ equal parts. For larger jobs, the low load average behavior is a linear increase in time-to-completion with load average but with an offset in elapsed time. This reflects the substantial waiting time (for a given job) to swap pages in and out. The I/O wait curve shows much dispersion at low load averages depending on the character of the system load. If there are not many jobs to be run, and one becomes unrunnable because of waiting for swapping or disk I/O, the wait time will be very high (see the upper curve of Figure 3). On the other hand, for the small, CPU-bound limit, the I/O wait loss is negligible (lower curve in Figure 3).

At load averages above 3 or 4, a non-linearity sets in for large jobs because of memory limitations and the increased swapping load (relative wait time, interrupt service, etc.) on the system. In the same limit the I/O wait approaches 15-20%. Of the 512 "core" memory pages currently on the system, almost 380 are available to users. With the current working set (5) limitation parameter (maximum 150 pages), 2-3 large jobs and up to 5 or so mixed jobs may be resident at once. If more than this number of jobs are runnable, some must be totally out of core and receiving no service. Because it is costly to move whole working sets in and out of memory (5-10 milliseconds per page), the system attempts to minimize "thrashing" by approximating a batch mode of scheduling, giving more run time before forcibly removing one program from memory to run another. This degradation is spread around evenly but causes added swap and core paging time in order to run a job and hence increases the per job time to complete.

However, based on user comments at loads of 3-5, the runtime increase because of load average (even without any additional overhead) becomes excessive as well - jobs that ran in several minutes at low load average begin to take tens of minutes at higher loads making interactions much more cumbersome. Another factor is that by the time there are more

(4) These bottlenecks refer to program execution. File storage has been another limiting factor for the system and is discussed later (see page 7)

(5) The working set is a group of pages which is a subset of the total active memory used by a program and which the system "guesses" (based on previous running history) will be addressed during the next running time quantum. In this way the system attempts to keep only those pages needed at any point for a program to execute during its time slice.

than 6 large jobs on the system, we begin to run out of drum swapping space. The current capacity is 3300 pages and, allowing for the monitor, can hold 5-6 full 256K work address spaces. Typically with a load average of 4-5, there are many more jobs on the system (25 - 30) with a range of memory requirements. Thus under such loads, the drum can accommodate even fewer large LISP jobs. Of course when the drum fills, swapping overflows to the much slower moving head disks and contends as well with regular disk I/O traffic. This substantially increases the system I/O wait time. As noted on page 12, we have implemented a page migration facility which assures that drum space is used only by active pages; but under heavy load we may still exceed the capacity.

From these data it is clear that with the typical mix of jobs on SUMEX-AIM including many large LISP jobs, above a load average of 5 or so the system runs out of memory, CPU, and swapping space at about the same time. Because of budgetary constraints we are not able to augment all three resources at once however. FOR A GIVEN LOAD, the effect of adding memory and swapping storage would be to linearize the response curve (Figure 2) through a reduction in system overhead at load averages above 4. For load averages in the range of 5-6, this could recover up to 15-20% in elapsed time/CPU minute. The augmentation of processor capacity to first order reduces the overall slope of the curves in Figure 2 and thus benefits users at all levels of loading. If the load is truly interactive (jobs complete or require terminal input after a few minutes), any speed-up in running time will tend to reduce the load average as well since the jobs will leave the run queue sooner. For many long, CPU-bound jobs this effect doesn't exist and the load average would stay the same - the overall run times for the jobs would be reduced however. In the ideal case, doubling processor capacity would improve elapsed time/CPU minute by 50%. This cannot be realized in fact since having a faster processor with the same memory means that the swapping rate will increase and hence total overhead will go up.

Because of the greater advantage for interactive jobs and because of budgetary considerations, the strategy approved by the AIM Executive Committee was to augment the CPU capacity as the first step, taking note of the certain need to augment memory and swapping storage soon thereafter. In addition to the technical arguments, the rationale for the decision also takes account of the fact that DEC CPU prices have been rising recently whereas memory prices are presumably still falling.

We examined both an upgrade from the KI-10 to a KL-10 and the installation of a dual processor KI-10. From a technical viewpoint, our preference was to upgrade to the KL-10, particularly in light of DEC's indication that the machine would be configured (microcode) within the year to efficiently run TENEX. For essentially economic reasons, however, the KL-10 option was not feasible. DEC marketing has taken a firm stand about selling KL-10's as "systems" which means that we would have had to upgrade not only the CPU but disks, tapes, and data line scanner as well. The net cost of upgrade would have been in excess of \$500,000 - well over our budget. In view of this and the feasibility of a dual processor system based on our studies, we decided to add a second KI-10. The implementation of this plan is now underway and proceeding very well - we are about ready to bring up the new system for user testing and hope to have it ready for use during the next AIM workshop this June.

It must be pointed out that at the time the decision for a dual processor was made (fall of 1975), we realized that the trend within the ARPA TENEX community could be toward a DEC-supported TENEX system although DEC had not made clear its plans for TENEX support. At this time that indeed seems to be the long term prognosis. DEC has announced the KL-20 (2040) running TOPS-20 which is a direct descendant of TENEX. The KL-20 is not a fast enough machine to have benefitted us in upgrade (it is slower than the KI-10) and delivery will only start in volume next fall. The rest of the KL-20 series of machine has not been announced although two bigger machines (currently denoted 1080T - to be called 20??) may be delivered to ARPA contractors late in 1976. A substantial amount of work remains to bring the DEC TENEX system up to the state of the current KA and KI TENEX systems which will likely take another year at least. Thus whereas in the long term the dual KI TENEX system will diverge increasingly from the DEC mainstream (by current projections), the pace with which it is coming into operational status and the minimal disruption to on-going user work, support the correctness of the decision relative to the pragmatic needs of the existing community.

We had expected delivery of the second KI-10 in late March but because of scheduling problems within DEC, we did not receive it until April 15. Because of additional delays in getting the needed memory and I/O bus cables to connect the new machine into the existing system, it could not be checked out to begin software development until the last week in April. Once installed, the machine has worked with only a few minor problems which were quickly corrected. Software development has gone equally well and we were ready to bring the full dual processor system up to begin user testing as of May 16.

Disk File Storage:

As mentioned earlier, the system has been operating at file system capacity as well over the past year. We have implemented policies which regularly clean out the file system (expunge deleted and temporary files as well as archive old files) to keep user projects within allocated limits. Nevertheless, many of the projects face severe constraints in available on-line storage needed for large LISP program development and community interactions. Because the system is fully allocated, there is little we can do to alleviate the problem within the present hardware. We are implementing operational improvements as possible to facilitate file archiving and restoring. We have also investigated the Datacomputer facility managed by the Computer Corporation of America over the ARPANET as well as other sources of on-line storage (at less loaded network sites) which could be available for a fee. The space available at the Datacomputer has been disappointing up to now. Some projects are trying out storing files at other sites but because of ARPANET access constraints, this is likely not a useful long term solution for the whole SUMEX-AIM community.

Since augmenting the file system is presently beyond our budget (higher priority is being given to improving computing capacity through CPU and memory enhancements), we have encouraged user projects which are

particularly cramped for space to assist in budgeting disk augmentations for themselves and the community. The DENDRAL and Chemical Synthesis projects each have proposals to cover additional RP-03 drives. We have two slots left on the existing controller and incrementally this is the lowest cost way to augment file space. Another highly attractive approach is using a System Concepts SA-10 "IBM" channel with double-density "3330" drives. However the initial cost of the channel, new controller, and one or two drives would be about \$100,000.

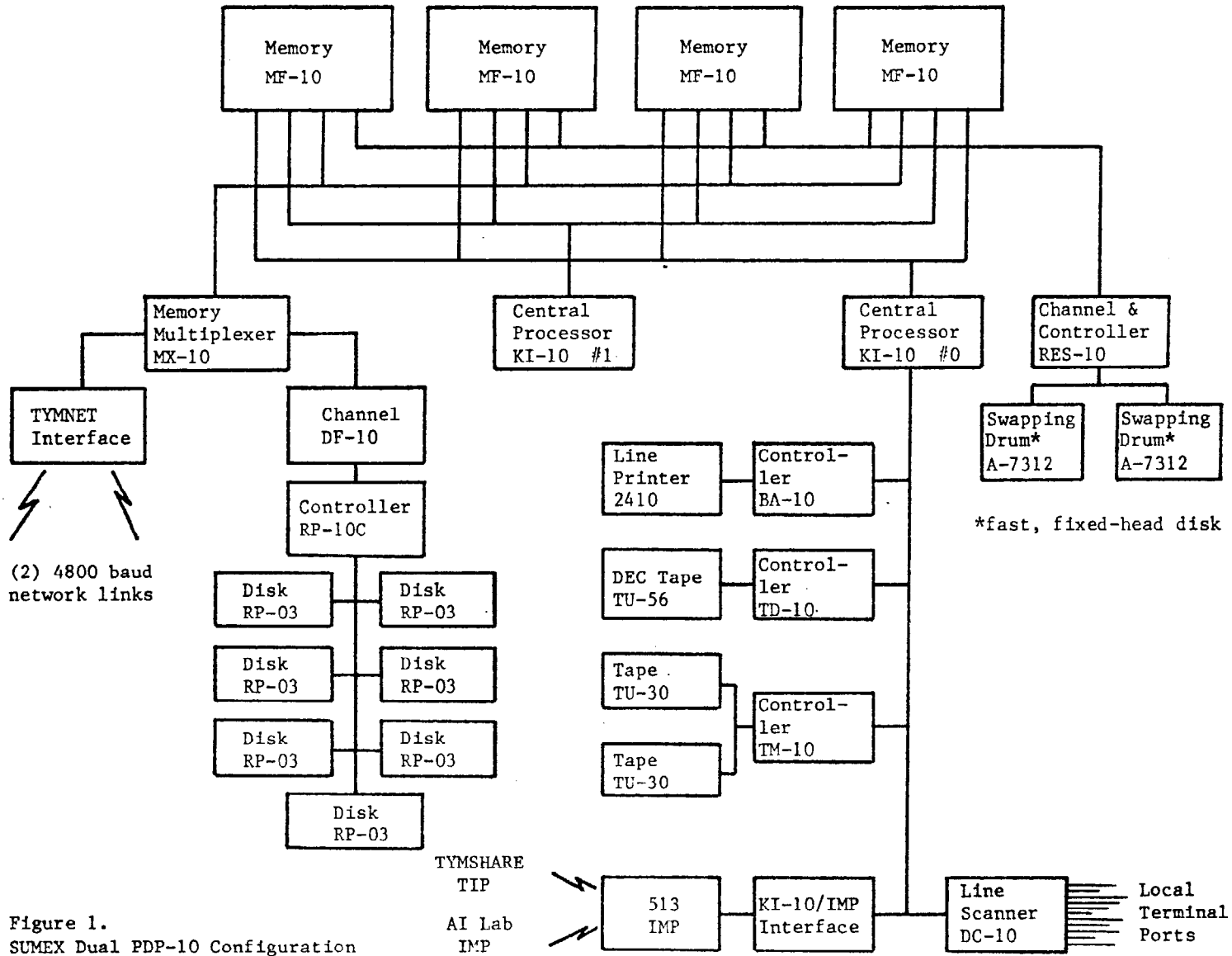


Figure 1.
SUMEX Dual PDP-10 Configuration

Figure 2.
SUMEX RESPONSIVENESS UNDER LOAD

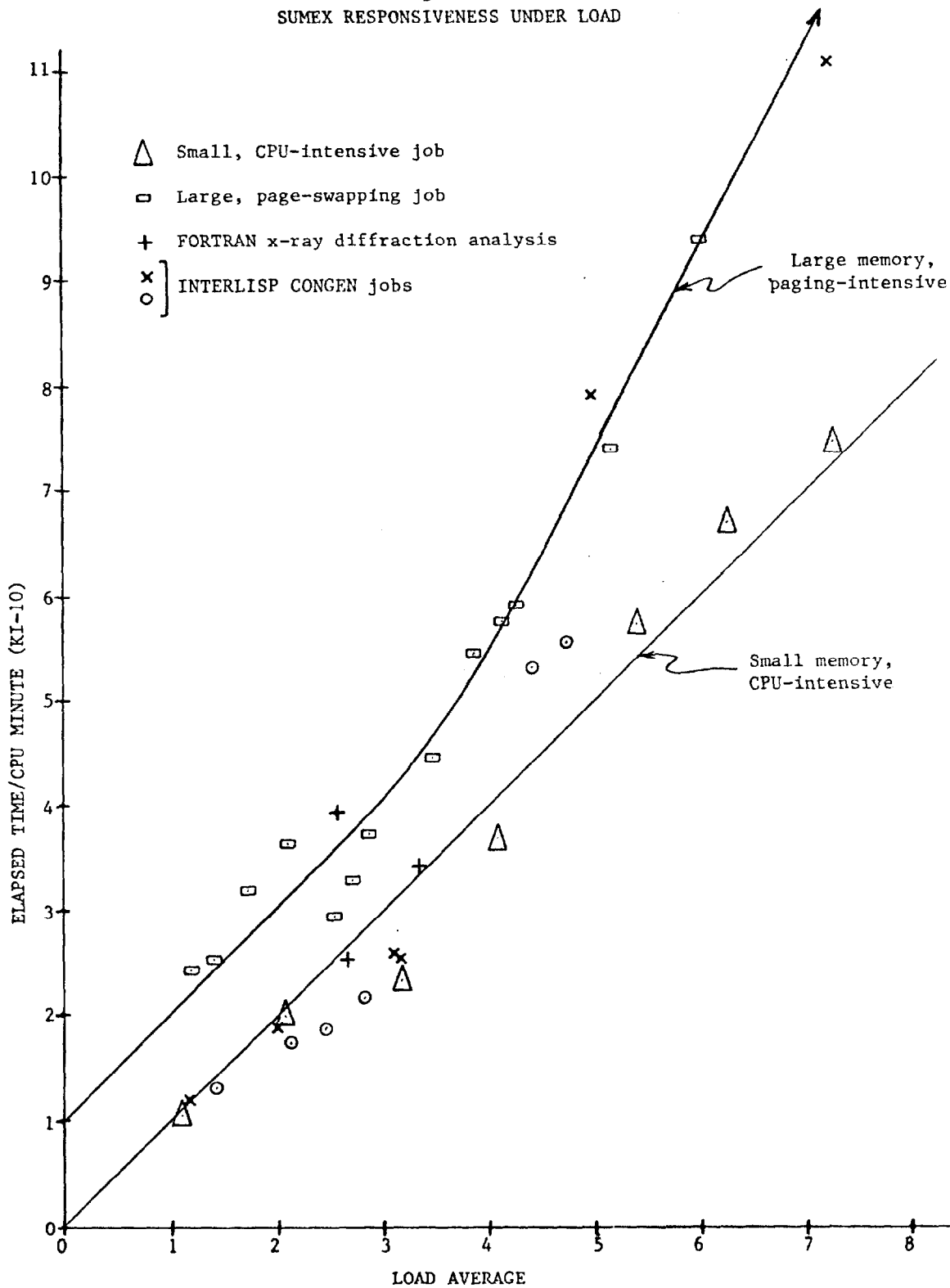
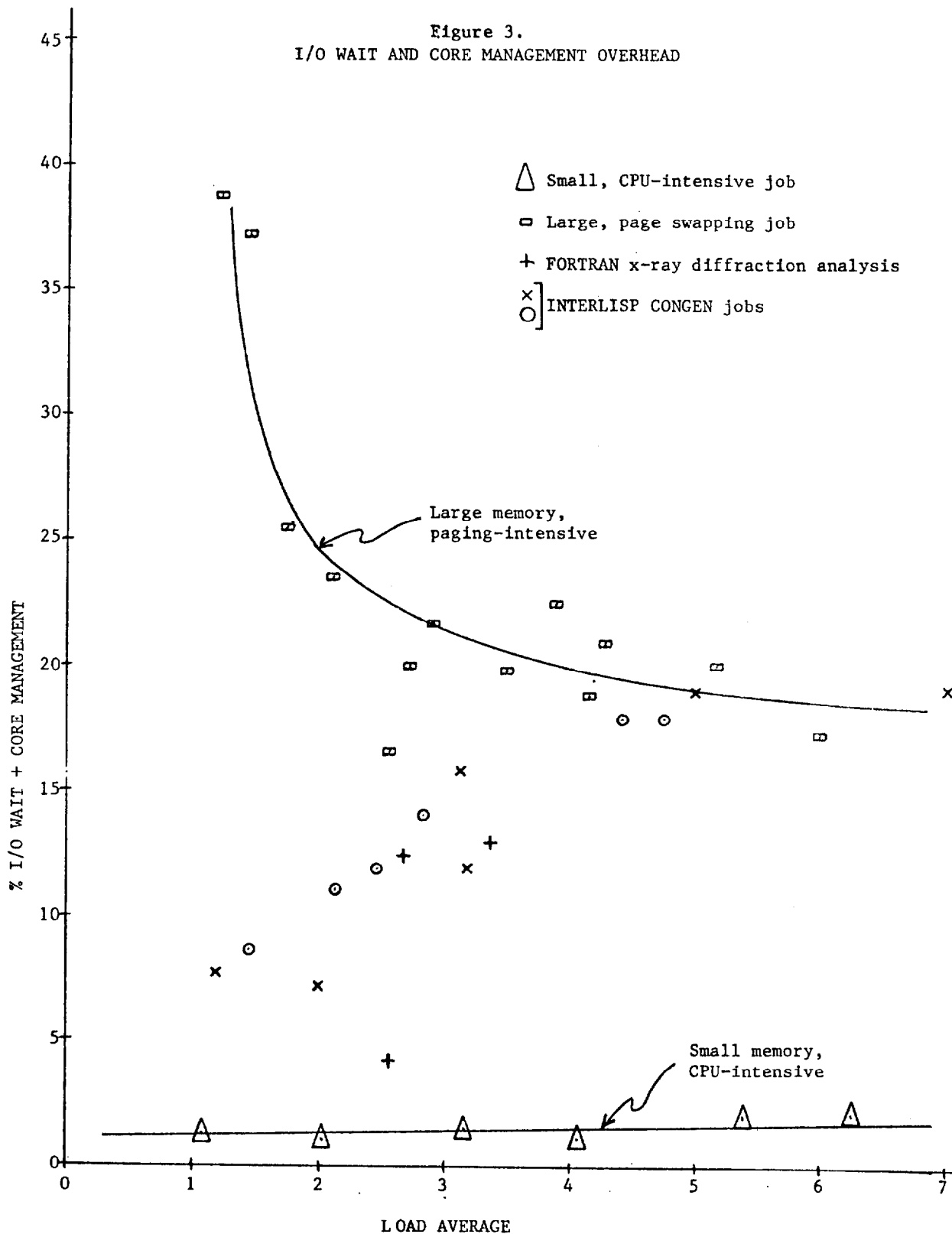


Figure 3.
I/O WAIT AND CORE MANAGEMENT OVERHEAD



II.A.2.b TENEX SYSTEM SOFTWARE

Our goals to improve resource allocation control capabilities, improve guest facilities, and maintain system compatibility with other ARPANET TENEX sites notwithstanding, we have continued to run TENEX release 1.31 this past year. The decision to upgrade to a more current release was deferred pending the decision on the processor augmentation. Had we been able to work out the acquisition of a KL-10, the monitor development effort would have had to take a different course to intercept DEC's monitor development efforts directly. Since that was not possible and a second KI-10 was approved in December 1975, we have begun the conversion to TENEX 1.33/1.34 in concert with the dual processor development effort (see below for a description of the trade-offs between the older 1.33 release and the very recent 1.34).

Swapping Storage Management:

Earlier in the year, while waiting for the processor decision, we finished implementing a drum page migration system which ensures that the drum is used only by active (recently accessed) pages. This optimizes the use of swapping space and reduces the substantial overhead when swapping overflows to the disk which is 5-10 times slower and contends with other file I/O. The garbage collector operates cyclically (currently every 10 minutes) and if a page allocated on drum has not been accessed during the previous interval and if alternative space is available on disk, it reassigns the page to disk. The cycle time of 10 minutes was chosen to give a reasonable time for a program to get around to using a page before declaring it dormant and at the same time not to penalize swapping of newly created pages by forcing them to reside on disk too long. This cycle time seems to be acceptable as we observe migrations of several hundred pages per cycle on the average. This has eliminated the situation where users first on the system leave dormant jobs around on drum and users who login later have their job pages allocated on disk because no drum space is free. Of course, during really peak loads the drum space may still become saturated with active jobs. We find that aggregate I/O wait times approach 40-50% with significant amounts of disk swapping whereas using drum, the I/O wait falls to under 20%.

Dual Processor Development:

Since the dual processor decision, the plan of attack has been: a) develop the dual processor system for KI-TENEX 1.31 which has been a thoroughly debugged system in our environment, b) in parallel, to transfer local TENEX changes (drum handler, TYMNET service, special JSYS's, etc.) to TENEX 1.33/1.34 and debug it as a single processor system, and c) after the dual processor system has stabilized and TENEX 1.33/1.34 is running well, complete the upgrade of our TENEX 1.33/1.34 to the dual processor configuration. The sequencing of these changes is designed to get the added capacity on line as soon as possible (particularly for the second workshop at Rutgers the first week in June) and to minimize the impact on users.

The dual processor software system is in the process of final implementation and checkout by R. Schulz and B. Hasselblad. We expect to bring the system up for user testing by mid May, conduct a detailed performance evaluation after the workshop in June, and complete documentation in July. Thus the following is only a preliminary report on the overall design philosophy. We will detail the system implementation and performance in the next report. From the start the design emphasized treating the two machines symmetrically and has maintained the ability to run the system either as a single or dual processor. The processors operate independently using common monitor code and system status data, each scheduling jobs independently, executing system calls, etc. The coordination between the machines is through status information in the data base and a set of interlocks which each machine can test and avoid simultaneous interference in sensitive areas. There have been many difficult issues in constructing the system of monitor interlocks and in debugging sections of monitor code for dual processor operation. This work has been greatly aided by the highly reentrant nature of the initial TENEX monitor design. The dual processor design has remained stable from its initial conception and implementation is proceeding on schedule. The detailed design began in early February with final design and implementation being done during late March and early April. After hardware installation during late April, debugging began. We began user access to the system on a test basis on May 16.

TENEX Monitor Upgrade:

Approximately 6 man-months of effort are being expended to upgrade the Tenex operating system at SUMEX from version 1.31 to 1.33/1.34. Version 1.32 was skipped because it was primarily a maintenance release which contained no new features or capabilities that we desired or required, although certain bug fixes and efficiency improvements were incorporated as deemed beneficial. We have been running version 1.31 for some 3 years. The major portion of work involves the incorporation of local SUMEX features into the new version including the dual processor changes, with the ensuing checkout and documentation phase.

Version 1.33 has been out in the field since January, 1975 and is a well proven and reliable system. It includes numerous bug fixes and improvements in efficiency along with a number of new features, the most important of which is the inception of the pie-slice scheduler. Version 1.34 is the most current version but has not been running as long. It has further updates to the pie-slice scheduler, bug fixes, and a reorganization of the source code. A final decision about whether to go to the proven 1.33 or immediately to 1.34 will be made before the end of May. We expect to have the new system up and running by late July.

The pie-slice scheduler provides system administrators with a mechanism for dividing user communities into groups ("pie-slices") and establishing minimum service levels for each group. These minimum service levels are guarantees which are met by the system regardless of activity in other groups. It is possible, of course, to observe a level of service in excess of the guarantee. This may happen either as a result of a group

being explicitly assigned the unclaimed share of an unrepresented group (the so-called "windfall") or simply as a result of small system load; no cycles are ever deliberately discarded. This represents a radical departure from the basically "laissez faire" 1.31 philosophy. In particular, at SUMEX where we have three somewhat separate user communities, a) local Stanford users, b) national AIM users, and c) SUMEX staff, we will be able to explicitly assign relative priorities of 40-40-20 respectively for the three groups and have the Tenex scheduler dynamically enforce them.

Completion of the 1.33/1.34 conversion effort has a few other implications. First, the dual processor software was developed as a parallel effort, so those changes will have to be incorporated into 1.33/1.34 as well. Second there is a new version of the EXEC (1.53) that goes along with 1.33 that has a number of new features and takes advantage of the new monitor JSYS calls provided by 1.33. Third there will be a reasonable documentation effort required, although most of the new features and commands are already documented in machine readable form, and only need to be put together in a suitable package. Fourth, there is the consideration of moving right on to Tenex 1.34. This version ties the core management functions of the operating system in more closely to the pie-slice scheduler, and would no doubt be beneficial in our environment.

In any case, the target is to have 1.33 (or 1.34) up and running long enough before September so that we can be sure we have a stable system, and any problems that arise can be ironed out during the summer.

TENEX Executive Upgrade:

Another area of software development is in the Executive program which is the basic user interface to manipulate files, directories, and devices; control job and terminal parameter settings; observe job and system status; and execute public and private programs. As mentioned above under "monitor", significant upgrade work here was delayed pending the decision on system augmentation since the TENEX upgrade affects the EXEC as well. As with all system work, we face a dilemma which is particularly strongly felt in this area; should we run a "standard" system or should we adapt things to user community needs and thereby tend toward a "home-brew" system? This is a difficult issue in that in many respects the SUMEX community is special - it includes a broad spectrum of users from professional computer scientists and programmers to biomedical research scientists and clinicians. The latter group, of course, want a minimum impedance to using the performance programs they are interested in while the former group wants a rich assortment of system facilities and as much flexibility as possible. Since most systems are designed for the programmer community, we have adopted the viewpoint that controlled augmentations of the system must be made to accommodate the medical user. Much of this work is still in process and will be for some time. The key point of this effort is to introduce knowledge about the individual user into the system (such as his usual defaults in using system functions, his level of expertise coupled to on-line assistance, his domain of interest to alert him to new information and perhaps personalized system commands

or macros convenient to his needs) so that he perceives a system tailored to his style and conventions in using the computer.

At this stage such information is stored and used from special files on a per program basis as in the MSG message reading program and TV editor. The EXEC has built in a number of parameter and subcommand setting commands which can be initialized by the LOGIN.CMD file. We will continue to devote effort in this area in up-coming work particularly to try to design a more uniform system pathway to such user-specific data.

Other EXEC command changes have been introduced to improve user interactions with the system (some developed locally and others designed by BB&N). These include commands for setting version retention specifications for files, purging (delete and expunge) individual files, improved system status displays, mail checking, TENEX error number interpretation, running programs explicitly as ephemerals (separate, transient address space) or non-ephemerals, and a group of SET and SHOW commands for various status conditions. One particular feature assists managing the large number of user written and supported programs that are available to the community. To keep these programs separate from the system-supported programs, another directory was created. Since the EXEC routinely searched only <SUBSYS>, <CONNECTED>, and <LOGIN> directories to find a program, it would miss all the user supported software. To solve this problem and give each user added control, we implemented a search path facility that is user settable. This allows each user to specify (with the SET PATH command) up to six directories (and their order) for the EXEC to search to find a program. The path is initialized by the EXEC to <SUBSYS>, <USESYS>, <CONNECTED>, and <LOGIN>.

Other changes will be forthcoming with the upgrade to TENEX 1.33/1.34 and the corresponding EXEC 1.53. These will include better ^C handling to solve type-ahead problems, a facility to have the EXEC periodically run a program for you (mail check, calendar check, etc.), system status displays accommodating the pie-slice scheduler, better human engineering in various areas, and a number of bug fixes.

System users can find up-to-date information on EXEC features through the EXEC manual and various on-line files:

```
<BULLETINS>NEW-EXEC.INFO
<DOC>TENEX-EXEC-MANUAL-UPDATE.INFO
<BULLETINS>LOGIN-CMD.BBD
<DOC>TENEX-133-EXEC.CHANGES
```

II.A.2.c NETWORK COMMUNICATION FACILITIES

A most crucial aspect of the SUMEX system is effective communication with remote users. In addition to the economic arguments for terminal access, networking offers other advantages for shared computing such as uniform user access to multiple machines and special purpose resources,

convenient file transfers for software sharing and multiple machine use, more effective backup, co-processing between remote machines, and improved inter-user communications. We have based our remote communication services on two networks - TYMNET and ARPANET. These were the only networks existing at the start of the project which allowed foreign host access. Since then, other commercial network systems (notably TELENET) have come into existence and are growing in coverage and services. The two networks to which we are currently connected complement each other; the TYMNET providing primarily terminal service with very broad geographical coverage and unrestricted user access, and the ARPANET having more limited access but providing a broader range of communication services. Together, these networks give a good view of the current strengths and weaknesses of this approach.

From the user's viewpoint, the reality of using a remote computer as if it were next door depends singularly on achieving the perception that a network connection is like a local telephone call to the computer. Current network terminal facilities do not quite accomplish the illusion of a local call. Data loss is not a problem in network communications - in fact with the more extensive error checking schemes, data integrity is much higher than for a long distance phone link. On the other hand, networking has as its underlying principle that through shared community use of telephone lines, widespread geographical coverage is possible at substantially reduced cost.

TYMNET:

Networks such as TYMNET are a complex interconnection of nodes and lines spanning the country (see Figure 4 on page 19). The primary cause of delay in passing a message through the network is the time to transfer a message from node to node and the scheduling of this traffic over multiplexed lines. This latter effect only becomes important in heavily loaded situations; the former is always present. Clearly from the user viewpoint, the best situation is to have as few nodes as possible between him and the host - this means many interconnecting lines through the network and correspondingly higher costs for the network manager. TENEX in some ways emphasizes this conflict more than other time-sharing systems because of the highly interactive nature of terminal handling (e.g., command and file name recognition and non-printing program commands as in text editors or INTERLISP). In such instances, individual characters must be seen by the host machine to determine the proper echo response in contrast to other systems where only "line at a time" commands are allowed. We have connected SUMEX to the TYMNET in two places as shown in Figure 4 so as to allow more direct access from different parts of the country. Nevertheless, based on delay time statistics collected over the past year from our TYMSTAT program, the response times are not very acceptable. The aggregate data are statistically summarized in Appendix D on page 195 and plots of the response time over the past year for particular nodes where we have extensive data are shown in Figure 5 on page 20. When delay times exceed 200-300 milliseconds, the character printing lag problems become noticeable with a full duplex, 30 char/sec terminal. These times have been particularly bad in New York with peak

delays approaching 3 seconds one way! Other nodes show uniformly high readings as well. These data reflect the subjective comments of many of our user groups as expressed in their individual reports (see Section IV on page 68). Problems have been particularly acute for Dr. Safir's group in New York and Dr. Amarel at Rutgers (see page 131).

We have had numerous meetings with TYMNET personnel to try to ease these problems and have instituted reroutings of the lines connecting SUMEX-AIM to the network. Also local lines to more strategic terminal nodes have been considered for users in areas poorly served by the existing line layout. These remedies have not had substantial effects. In general the TYMNET design goals are not to provide much better service. To quote from the April 1, 1976 TYMNET User's Group Newsletter:

"Current delay experience is from 0.25 to 3 seconds for a character to make a round trip through the network, with an average of 1.2 seconds. By early next year, "Clusters" will be installed in high density areas and will be interconnected by 9600 BPS lines. The result will be that round trip response/delay time will be less than 1 second for 80% of the cases and less than 2 seconds for 98% of the cases. This also is the design objective for TYMNET II."

We will continue to pursue improvements in TYMNET response but user terminal interactions such as used in TENEX programs are not realized in the time-sharing systems offered by most other TYMNET users and hence are not supported well by TYMNET. With these delays, it is not clear how well the proposed 1200 baud service they are going to inaugurate will work.

ARPANET:

The ARPANET, while designed for more general information transfer than purely terminal handling, has similar bottleneck problems in its topology (see the current geographical and logical maps of the ARPANET in Figure 6 and Figure 7 on page 21). These are reduced by the use of relatively higher speed interconnection lines (50 K baud instead of 2400 - 9600 baud lines as in TYMNET) but response delays through many nodes become objectionable eventually as well.

We are enforcing a policy to restrict the use of the ARPANET to users who have affiliations with ARPA-supported contractors and system/software interchange with cooperating TENEX sites. The administration of the network passed from the ARPA Information Processing Techniques Office to the Defense Communications Agency as of July 1975. At that time policies were announced restricting access to DoD-affiliated users. We have protected the facilities for calling from SUMEX out to other sites on the ARPANET to authorized users. This also protects the SUMEX-AIM machine from acting as an expensive terminal handler for other

machines - this function is better fulfilled by dedicated terminal handling machines (TIPS). In general, we have developed excellent working relationships with other sites on the ARPANET for system backup and software interchange - such day-to-day working interactions with remote facilities would not be possible without the integrated file transfer, communication, and terminal handling capabilities unique to the ARPANET.

We take very seriously the responsibility to provide effective communication capabilities to SUMEX-AIM users and are continuously looking for ways to improve our existing facilities as well as investigate alternatives becoming available. We are investigating the TELENET facilities that have been rapidly expanding this past year. BB&N has hooked one of their TENEX systems up to TELENET and subjective reports are that response problems similar to those reported above were present there as well. We have requested specific data on their experience but have not received any yet. We have received comments particularly from Professor Colby's group which uses the ARPANET primarily (see page 116) that serious network delays place the remote user at a substantial disadvantage in competing for system resources and that compensating biases in allocation procedures should be implemented to offset the problems. Another critical problem is the lack of high speed printing facilities local to remote groups. The new system being installed should help assure remote users their fair share of CPU; but a simple bias in system percentage will not offset network delay problems. The communication problems must be solved as communications problems and the only way to ensure good terminal response is to provide high enough speed lines that are not over loaded.

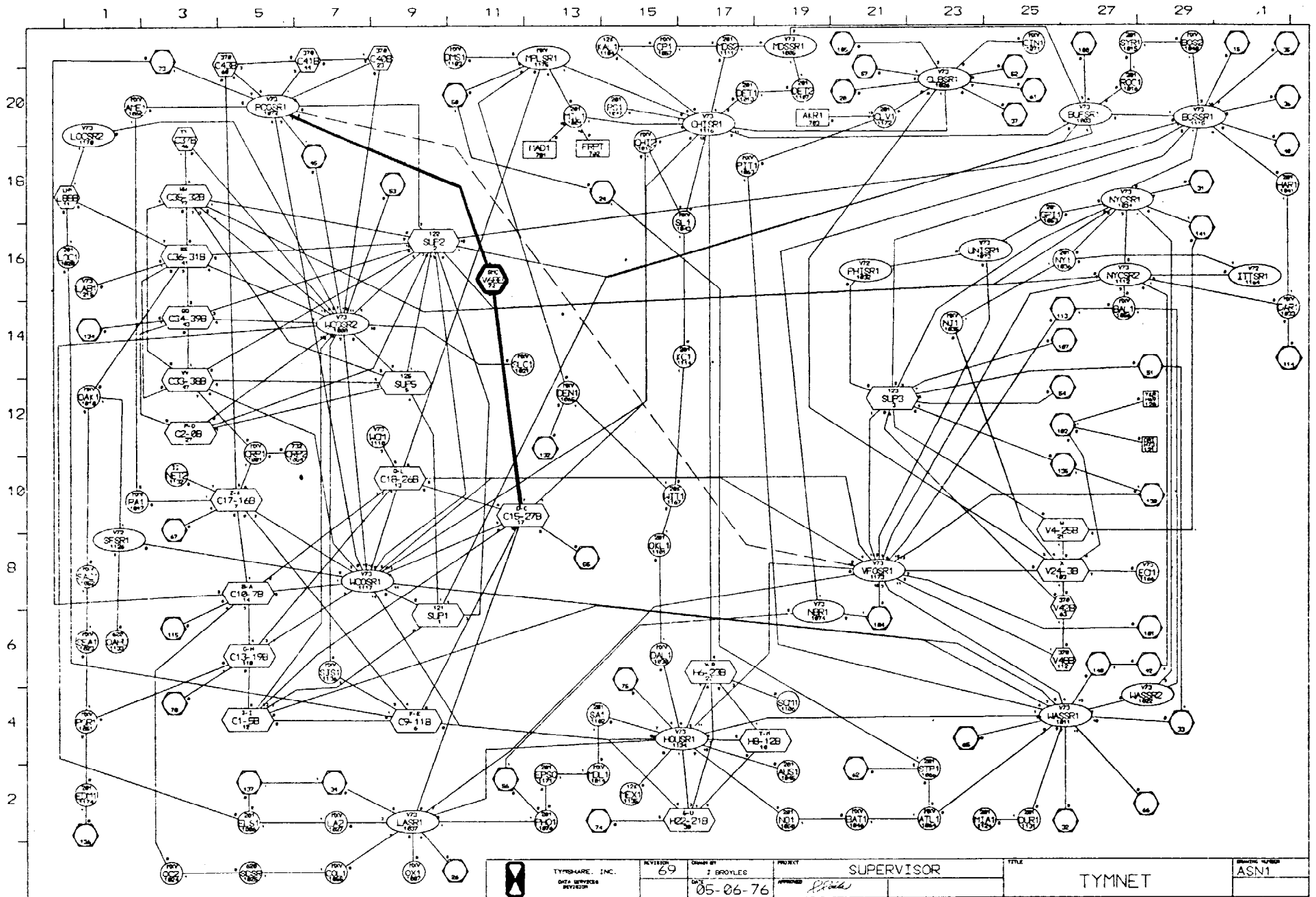


Figure 4. TYMNET Network Map

Figure 5.
TYMNET RESPONSE DELAY STATISTICS

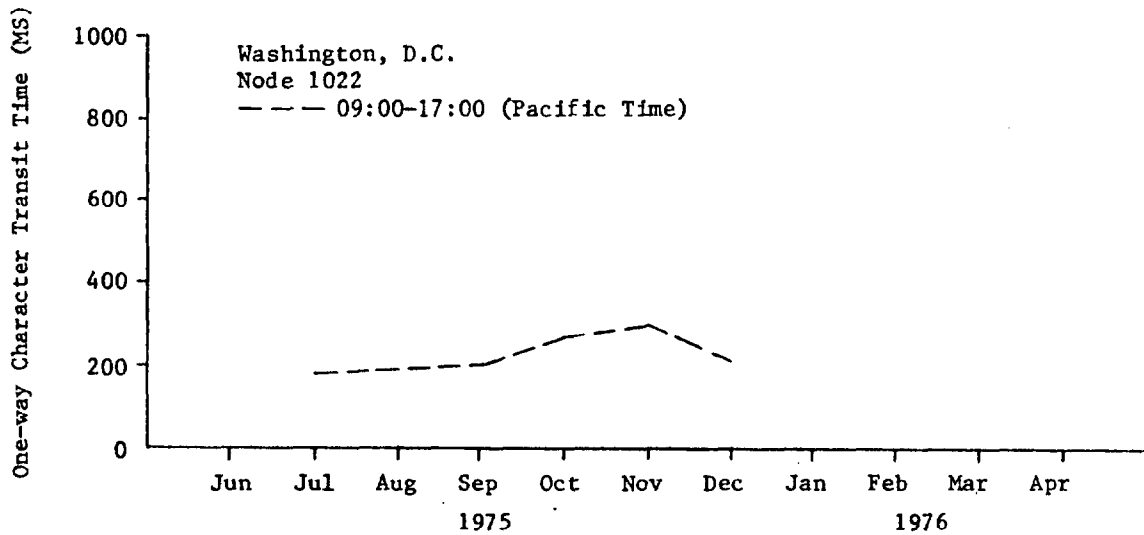
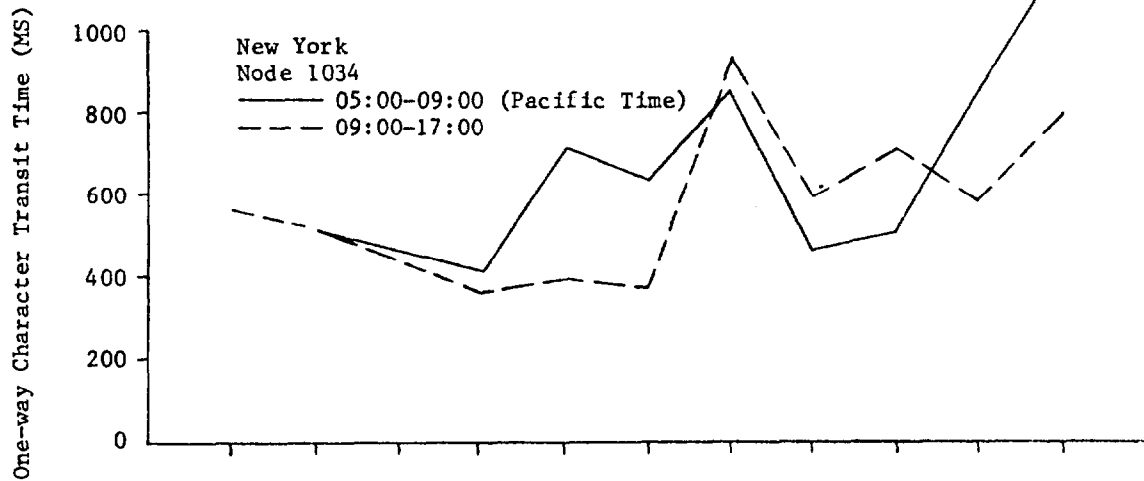
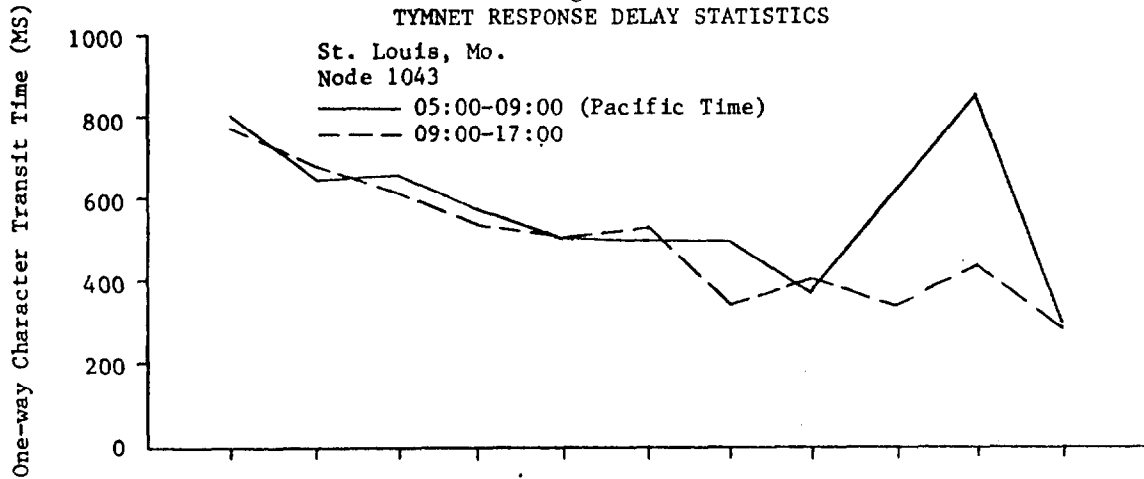


Figure 6.

ARPANET GEOGRAPHIC MAP, FEBRUARY 1976

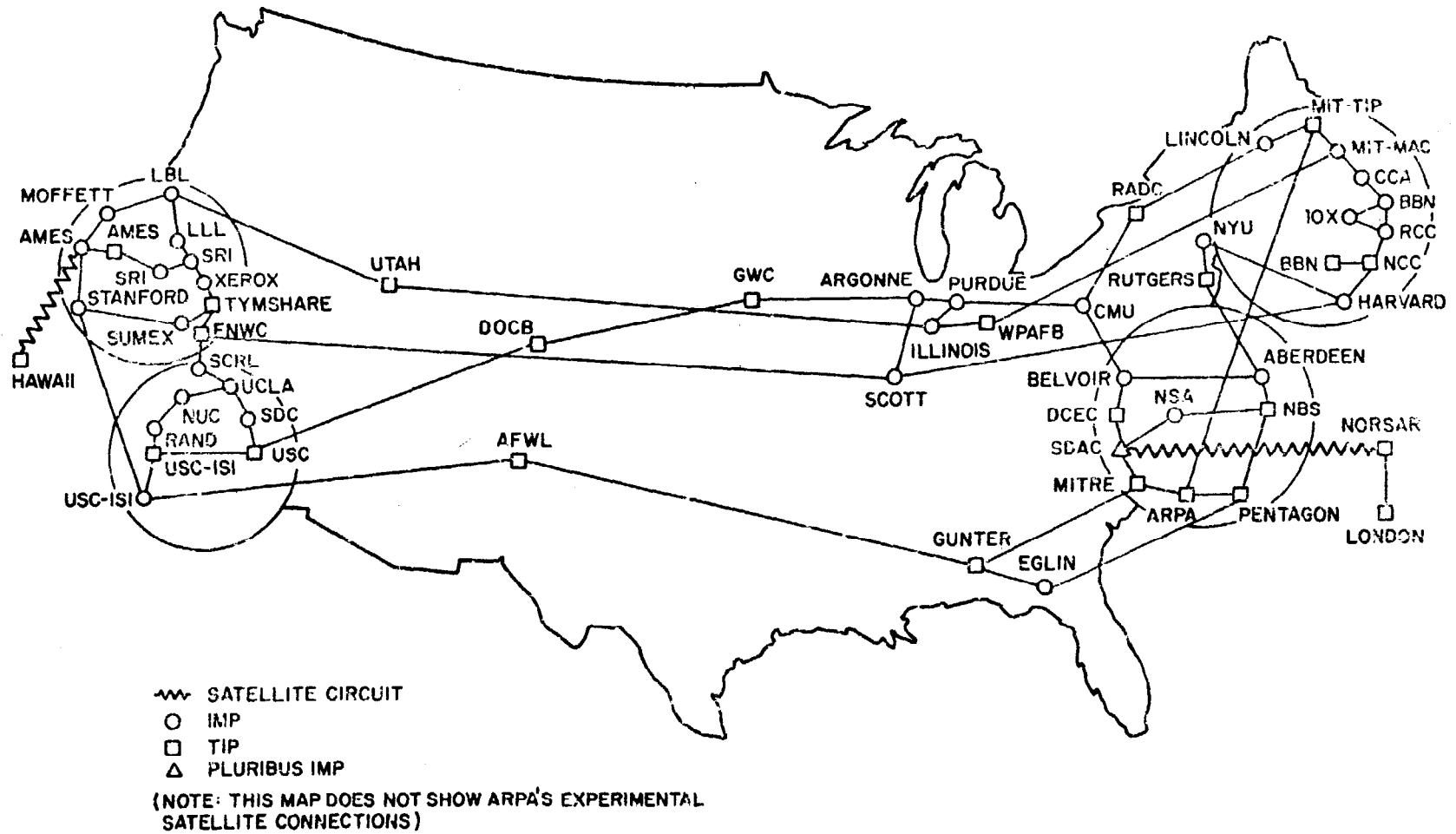
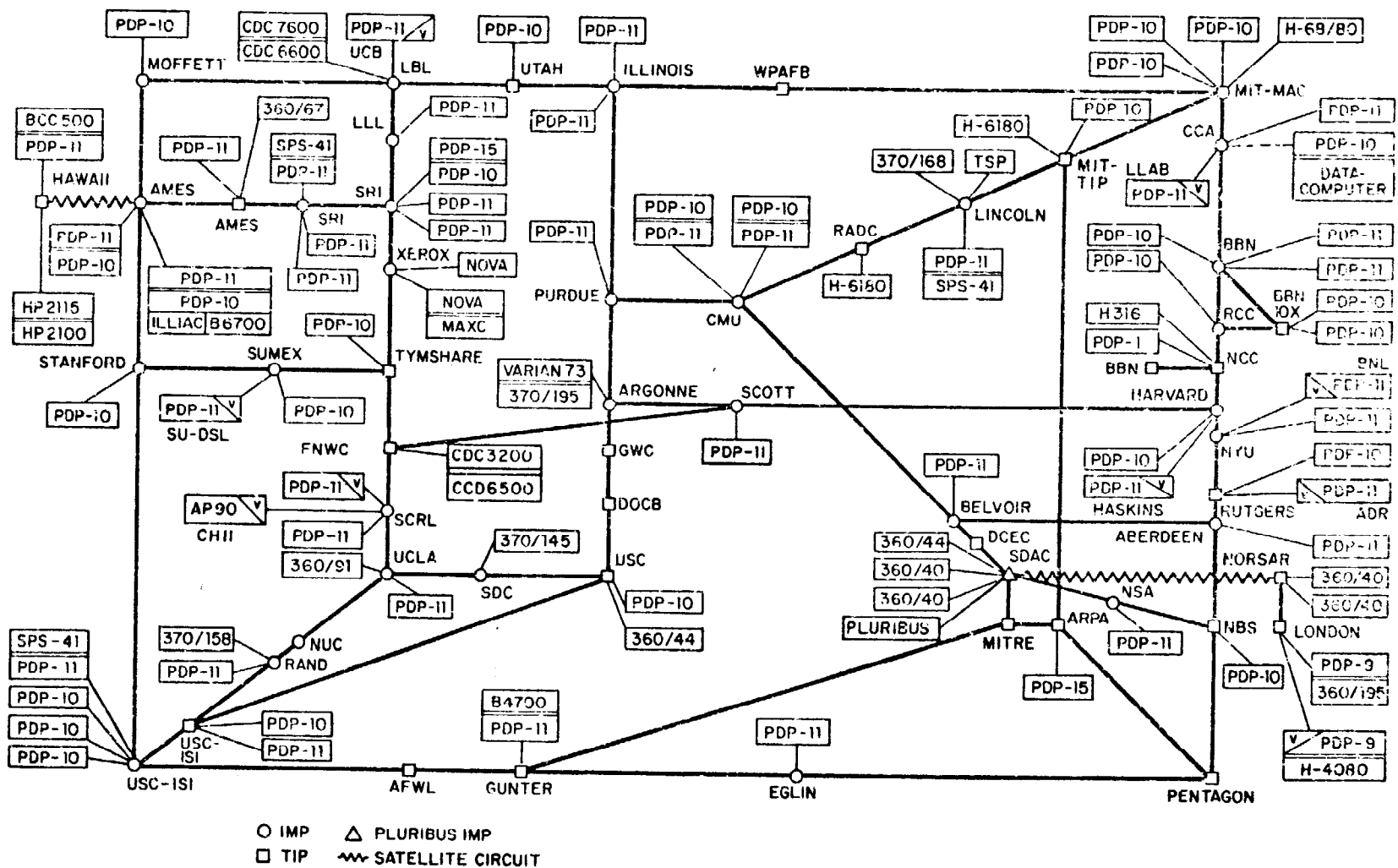


Figure 7.
 ARPANET LOGICAL MAP, FEBRUARY 1976



(PLEASE NOTE THAT WHILE THIS MAP SHOWS THE HOST POPULATION OF THE NETWORK ACCORDING TO THE BEST INFORMATION OBTAINABLE, NO CLAIM CAN BE MADE FOR ITS ACCURACY)

II.A.2.d SYSTEM RELIABILITY AND BACKUP

System reliability has improved over the past year; excellent under stable hardware and software conditions and degrading during debugging and development periods (drum debugging, dual processor work, etc.) and during periods of hardware problems. The pertinent data are given below with indications of periods during which development took place.

SUMEX-AIM CRASH FREQUENCY (crashes/month)
AND DOWN-TIME DATA (hours/month)

Crash Type	1975						1976					
	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	JAN	FEB	MAR	APR
DEC HARDWARE	10	7	22	26	8	10	4	16	9	6	1	6
SOFTWARE	4	3	6	6	6	0	3	5	1	3	2	5
ENVIRONMENT	0	0	1	0	0	0	1	0	1	1	2	1
TYMNET HDWRE	0	0	0	0	1	0	0	0	0	1	1	0
UNKNOWN	1	0	1	0	0	0	1	0	1	1	0	0
DOWN-TIME												
SCHEDULED	80	79	98	123	72	52	52	43	41	48	38	67
UNSCHED	28	19	30	42	7	4	3	21	26	11	2	7

DEFINITIONS:

Crash = Any occasion on which an operational system must be restarted or reloaded. Multiple crashes while trying to reload are not counted unless the system comes up fully between crashes.

DEC Hardware Crash = Any crash caused by a failure in the PDP-10 hardware or peripheral equipment (CPU, disk, drum, etc.)

Software Crash = Any crash caused by a malfunction within the TENEX software system.

Environmental Crash = Any crash caused by power failure, air conditioning outage, lightning, etc.

TYMNET Hardware Crash = Any crash caused by the TYMNET hardware or the interface to the PDP-10. This includes only the times when a TYMNET problem causes the PDP-10 to crash and not the times when the TYMNET goes down and the PDP-10 continues in operation.

Unknown Crash = All other crashes in which the cause is not assignable.

Scheduled Down-time = Preventive maintenance time (6-8 hours/week), file system backup (3-6 hours/week), scheduled maintenance to repair non-critical component failures, and system development activities requiring a stand-alone machine.

Unscheduled Down-time = Time lost because of unexpected hardware or software failure. For the most part this is the time to diagnose and either repair the problem or to reconfigure the system and bring it up to run in a somewhat degraded mode until a later scheduled shutdown for permanent repair.

Whenever development efforts are undertaken which affect the system hardware or monitor, additional downtime and some period of unreliability may result causing more crashes than are representative of the overall reliability of the system. The following gives some insight into these development efforts as reflected in the above data.

Jul - Sep 1975: Debug drum system error rate problem.

Late Apr 1976: Begin dual processor installation.

As can be seen, we have had some periods of hardware unreliability stemming mostly from intermittent problems. Particularly troublesome components of the system in terms of such problems have been the disk drives, memories, and during hardware relocations, the inter-device cabling. The KI-10 CPU has been very stable and given only one problem over the past year (an I/O bus driver).

From the user's viewpoint, besides the obvious inconvenience of not being able to work during down time, the fragility of the highly interlinked TENEX file system has caused only a few occasions of having to backup to previous file system states this past year. We save changed files daily and copy the entire file system to fresh disk packs weekly. Thus an unexpected crash may cause the loss of up to one day's worth of work - it in fact may take longer for a given user to reconstruct the lost work if complex debugging or development changes were involved and undocumented. When the system is known to be subject to intermittent crashes, we backup more often to protect users.

Our current schedule for system backup is early Sunday morning (Pacific Time). We now have two students who do the file system backups at night as well as the archive/retrieve requests. By moving these activities to night hours, we off-load them from the prime time and also provide added coverage for quick recovery from any system crashes. This does not require full time attention and the students also help out with system programming tasks in developing utilities.

Another aspect of reliability and backup is the need to assure computing service for critical demonstrations, lectures, and the like. We have a good mutual relationship with existing ARPANET TENEX sites for such backup when needed (e.g., for the AIM workshop).

II.A.2.e PROGRAMMING LANGUAGES

Over the past year we or members of the SUMEX-AIM community have continued to maintain the major languages on the system at current release levels, have TENEXized several languages to improve efficiency, and have investigated a number of issues related to the efficiency of programs written in various LISP implementations and the exportability of programs. These issues are becoming increasingly critical in dealing with AI performance programs which have reached a level of maturity so that substantial, non-developmental user communities are growing. The following summarizes general accomplishments and the following section discusses in detail the work this past year in designing a machine-independent SAIL system (MAINSAIL).

General Language Support:

The ALGOL-like modeling language, SIMULA, was requested by the DENDRAL group for consideration as a language in which to implement a more efficient version of the chemical structure generation programs. The most recent release of SIMULA has been brought up on the system. It is also used by a number of the Rutgers project members.

Two existing programming languages were TENEXized by Mr. Tom Wolpert of IMSSS. TNXFAIL is now the official version of FAIL for TENEX sites. His code has been incorporated under compilation switch into the standard FAIL sources maintained at the Stanford Artificial Intelligence Laboratory (SU-AI). Mr. Wolpert also TENEXized UC Irvine - LISP (ILISP) which is an extension of LISP 1.6 to include the break package and editor facilities of INTERLISP circa 1971. ILISP is used extensively by Prof. Colby's group at SUMEX.

The latest DEC release of FORTRAN10 was installed late last year and is relatively stable although several bug fixes have since been made. As part of an effort to remain current with all new DEC releases, we have also updated the versions of: MACRO, BLIS10, and BASIC.

Two other languages which received active maintenance at SUMEX this year are INTERLISP and SAIL. New versions of INTERLISP are continually being issued by XEROX-PARC and are brought up on SUMEX by Mr. Larry Masinter (of Xerox) and Ms. Suzanne Johnson. Because of the large number of LISP programs that are written in various versions of INTERLISP (which are not necessarily compatible with the new releases) and the need to keep

many of these programs running for the growing community of collaborative users, we have implemented orderly scheme of introducing the new versions slowly. Old versions are removed only when there are no longer any SYSOUT's on them. At the same time we actively encourage users to keep their programs up-to-date to minimize the maintenance problems with LISP versions no longer supported.

TENEX-SAIL is maintained by Dr. Robert Smith of IMSSS/SUMEX and is exported through SUMEX so excellent service is locally available. A PRINT feature was added to SAIL this year in response to user suggestions as well as a number of new runtime routines. This year the RECORD data structure became available in SAIL and an effort has been made to familiarize users with this structure which is well-suited to AI applications. A collection of utilities for SAIL programming has also been gathered at SUMEX and introduced to the SAIL users.

A very important new part of the SAIL system is the BAIL interactive debugger which was written by Mr. John Reiser of SU-AI but in close consultation with SUMEX/IMSSS using the TENEX facilities to test the BAIL/TENEX interaction. BAIL allows users to interactively examine and change the contents of previously-defined variables and to enter SAIL statements using a subset of the language. Mr. Reiser offered a BAIL class at Stanford to introduce his system to local users.

LISP Efficiency:

There has been an on-going debate this past year between advocates of INTERLISP and ILISP over the relative efficiencies of the two languages and the level of assistance the language systems provide the user in developing programs. These issues are important because they influence the time required to develop new AI programs and subsequently the incremental load placed on the SUMEX machine when in use. A number of people have contributed to an evaluation of these two LISP's including Dr. R. Smith (IMSSS), Dr. Tom Wolpert (formerly of IMSSS), Mr. Larry Masinter (Xerox PARC), and Mr. Larry Fagan (MYCIN project - formerly USC-ISI). The tests were based on an implementation of a subset of REDUCE (a symbolic algebra manipulator). The results of several iterations in program refinement by experts in the respective languages were that the runtimes for the two versions were quite comparable (far less than the 5-10 disparity predicted by ILISP enthusiasts). A more disquieting result was the substantial difference in runtimes depending on how particular functions were coded IN THE SAME LANGUAGE. It is apparent from the results that factors of 10 differences in time can result from a superficial implementation - expert programming insight is essential to efficient program performance. This is not a real surprise in that it is true of programming in any language - the problems may be increased by such a rich language as INTERLISP with such a wide array of ways to do the same thing but with little guidance as to the relative costs. It has proven very difficult to quantify the "rules" for good programming. Mr. Masinter and Mr. Phil Jackson attempted to document good INTERLISP programming habits and issued a bulletin for SUMEX users.

A further impact of these data is that it is very difficult to simultaneously develop a new AI program and make the implementation highly efficient. With the iterations required to develop the conceptual design of the program, it is difficult to ensure its efficiency. This may lead to the need to reimplement the program after the basic development stabilizes to increase efficiency while still accommodating convenient and orderly further development. Such reimplementation may or may not be best done in LISP - this will depend on many factors including the nature of the program data structure requirements and anticipated further development efforts.

II.A.2.f MAINSAIL OVERVIEW

Another aspect of SUMEX's role in encouraging community software sharing which has received substantial attention this past year is the set of problems involved in software exportation. The following is a general description of the on-going development of a machine-independent programming system (MAINSAIL) by Mr. Clark Wilcox of our staff. A more detailed description of the language elements can be found in Appendix E on page 208.

The MAINSAIL programming system (referred to as SAILEX in the 1975 annual report) has undergone extensive development during the past year. The considerable interest expressed to date from across the country indicates that MAINSAIL could be a significant step towards the distribution of portable software (programs which can be executed, without alteration, on a variety of computer systems). In response, SUMEX is pursuing plans to make MAINSAIL available to other sites, and to promote the exchange of programs within a diverse computer community. This type of language and program sharing is now made more difficult by the incompatibilities among the various implementations of current languages. MAINSAIL embodies a unified approach which presents to the user the same programming system, regardless of what computer or operating system supports its execution.

SUMEX, in its role as a nationally shared computer resource, is an appropriate vehicle for the development of high-quality software unbound by the underlying machine environment. We have a built-in community of program developers acutely aware of the significance of providing their work to a broader base of users. This intersection of hardware capability, software expertise, and dedication to resource sharing presents a unique opportunity to promote a system designed for program sharing.

MAINSAIL is being developed for two closely-related reasons: as a general-purpose programming system, and as a tool for research into the design of a machine-independent programming system. MAINSAIL will be fully implemented and actively used on a number of machines. It is perhaps one of the most highly-developed languages available for the mini-

computer environment. Its machine-independent design allows it to be used for the development of programs on one computer which will be executed on another. This capability will be of increasing importance as smaller dedicated computers are used in conjunction with larger general-purpose computers, and as programs are more readily exchanged over computer networks.

The MAINSAIL language is derived from SAIL, a programming language developed at Stanford University's Artificial Intelligence Laboratory. It is not compatible with SAIL, since SAIL was designed for a PDP-10 with TOPS-10, and hence contains machine-dependencies. However it has retained the basic attributes of SAIL as an extended ALGOL-like language. Among MAINSAIL's language features are:

- machine-independent language design
- straightforward syntax and semantics
- efficient code generation for variety of machines
- separately compiled segments
- double precision integer and real
- bit and address manipulation
- variable-length strings
- dynamic and static records
- generic procedures
- default and repeatable arguments
- static initialization
- in-line assembly language
- macro facility
- compile-time evaluation
- conditional compilation
- multiple source files during compilation
- sequential and random i/o
- terminal interaction
- comprehensive system procedures
- mathematical library
- access to dynamic storage allocation
- access to runtime system
- "garbage collection" of strings and records.

MAINSAIL is designed as a programming system rather than just a programming language. It is presently composed of a compiler generator, a compiler, and a runtime system. Further components envisioned are a debugger, a code optimizer, and a text editor. All of these components are to be written in MAINSAIL, and hence made fully portable. Also, there are plans for extensions to the MAINSAIL language, such as:

- coroutines
- extensible data types
- extensible code generation
- list processing system (LEAP).

In its role as a research tool, MAINSAIL is being used for an examination of the following design issues:

- language design --- what syntactic and semantic constraints are imposed by portability.
- compiler design --- how can a single language be compiled into efficient code for a large number of computers.
- runtime design --- to what extent can the runtime system be made transportable.
- computer design --- what architectures best support a high-level language implementation.
- program design --- what role can portability play in the design of reliable software.

Since the PDP-10 and PDP-11 implementations will be the first to become available, this user community has been introduced to MAINSAIL at several meetings of DECUS, the Digital Equipment Corporation Users Society. MAINSAIL was first described in a talk delivered at the DECUS DECsystem10 Fall '75 symposium in Los Angeles. It was then featured in a session entitled "Languages for Portability", at the DECUS DECsystem10 Spring '76 symposium in Hyannis Port. A paper will be presented at the DECUS mini/midi Spring '76 symposium in Atlanta. These sessions are resulting in an almost continual stream of inquiries concerning MAINSAIL.

Before MAINSAIL is exported to other sites, it will be thoroughly tested on several local computer systems. It is now being used on a PDP-10 with TENEX and a PDP-11 with RT11. Implementations for a PDP-10 with TOPS-10 and an IBM-370 with ORVYL (Stanford operating system) are now under development. Code has also been generated for an INTERDATA-716 and a Data General NOVA, both mini-computers. There is interest in using MAINSAIL on a PDP-11 with operating systems such as UNIX and RSX-11; a DECsystem20; and the NIH-GPP, a parallel processor being constructed for the NIH Image Processing Unit. MAINSAIL will be made available on such machines as sufficient funding is obtained to support an expanded effort.

A number of projects are interested in using MAINSAIL for the development of portable software. Among these are:

- robotics project
- mass spectrometry system
- computer-aided-instruction system for the teaching of logic
- automated cell classification laboratory
- machine-independent version of (a subset of) INTERLISP
- display-oriented text editor (TV-EDIT)

Several INTERLISP programs are being considered for translation into MAINSAIL. These programs have developed to the point that they no longer require the very general environment supported by INTERLISP, and hence can avoid the related inefficiencies. Also, there is interest in providing such programs to a wider community with systems other than a PDP-10 with TENEX.

MAINSAIL can furnish the capabilities necessary to develop generally useful software, and to distribute that software to a wider community than any existing language can now reach. To reach that goal, several issues must be resolved.

First, it must be demonstrated that MAINSAIL can be efficiently implemented on a wide variety of computer systems. The creation of new implementations must be largely automated, with the machine-dependent aspects requiring at most a few man-months for development. Some plausible target systems are:

PDP-10 (TENEX, TOPS-10)
 PDP-11 (RT-11, RSX-11, RSTS, UNIX)
 DECsystem20
 IBM-360/370 (ORVYL, TSS, TSO)
 NOVA/ECLIPSE
 additional large machines: CDC, UNIVAC, ...
 additional small machines: INTERDATA, TI990, MICRODATA, HP-3000, ...

Second, a suitable means of distributing the MAINSAIL programming system, along with portable software written in MAINSAIL, must be determined which is consistent with resources available to SUMEX. Steps must be taken to insure compatibility among the various sites, and to minimize maintenance of the machine-independent parts. The development and distribution of MAINSAIL is viewed as a time-consuming process fraught with complications. SUMEX must be careful not to make promises which cannot be kept.

Finally, program developers must be motivated to design machine-independent systems, to promote their distribution, and to use portable software developed at other sites. Machine-dependencies must be eliminated, or at least isolated and documented. Program design must face, from the onset, the restrictions imposed by portability.

II.A.2.g OPERATIONS AND USER SOFTWARE

Operations Programs:

The programs which assist system operations and management have been effectively organized this past year. A catalog has been made of approximately 40 operator programs which had been maintained by various staff members to facilitate their particular tasks and passed along to other staff members informally only. Some of the purposes of the new operator utilities include: setting the GUEST password and the access of guests to other user programs (only restricted access is allowed for guests); providing summary information on directories and groups;

exercising various hardware devices; performing error analysis for system crashes; changing system downtime information; creating new directories and setting the appropriate parameters for each user; sending important notifications to logged in terminals; watching terminal activity; and handling special resource allocation situations such as project demonstrations. Many operator tasks have also been automated as "autojobs" or batch jobs which run on the system performing tasks continuously or on predefined time schedules. In addition to this collection of basic operator utility programs, software has also been improved for system-wide statistics gathering, accounting (including information on diurnal community loading and service), disk allocation checking and enforcement, system backup onto tape for file protection (any file in existence for 24 hours will now be guaranteed retrievable for a two-month period), and currently a new spooler is being written for the lineprinter to provide faster handling with more control of the printer queue available. Another development effort is in the area of collecting and organizing statistics about subsystem use. Such data can help in planning where to concentrate development effort, determining which users to notify about new program information, and getting users and user groups with similar interests together.

User Software:

As the user community has stabilized certain trends in user needs have been noted and utility software has been collected and written in those areas. Particular attention was also paid this year to making the user interfaces for all existing utilities as consistent as possible and many programs received minor cosmetic modifications.

STATISTICS PACKAGES

Three new utilities packages were added: STP (STATPACK Statistical Package) and BANK (a data management package) both from Western Michigan University; and SPSS (Statistical Package for the Social Sciences) converted by University of Pittsburgh for PDP10 use (and obtained from them under contract).

DEC SOFTWARE

There is a new DEC policy to issue maintenance releases at least every two years for all software. This has resulted in a large volume of new DEC releases over the past year. Essentially we receive DEC software under three separate arrangements. First, FORTRAN10 and LINK10 were purchased under contract. Second, we have subscribed on an annual basis to the DEC distribution tapes from which this year, we put up new versions of RUNOFF, PIP, BASIC, BLIS10, CCL, FILEX, CREF, CHANGE (known at SUMEX as DCHANGE due to name conflict), FLECS, GLOB, MTCOPY, MACRO, TECO, and LOADER. A variety of bug fixes were done on these programs and sent back to DEC. Also, the SCAN program was modified for handling TENEX-style directory names for these programs. For most of the programs, after some practice and with the help of a TECO routine to automatically check the stored SRCCOM record of local modifications to the old versions, it became

a one or two day effort to get each program up. The third source of DEC software is from the DECUS tapes which are ordered individually with a minimal per tape charge. We order all programs from these tapes which appeared to be possibly useful. The tapes are then stored until a user request is actually received for a given program. The only such request to date has been for SIMULA.

PA1050 EMULATOR

All of the above official DEC software plus an assortment of other programs imported from TOPS10 sites can only be run through the use of a TOPS10 emulator - a program which converts all TOPS-10 system calls (UUO's) into "equivalent" JSYS TENEX calls. The PA1050 emulator is a standard part of TENEX software. However, it is badly out-of-date and inadequate for many complicated programming situations. SUMEX has developed its own version of PA1050 (incorporating features of other local versions wherever possible). The SUMEX PA1050 has now received reasonably wide distribution to other TENEX sites. Modifications to PA1050 this year included: conditional code for SRI and IMSSS use of PA1050, a change in the standard altmode character, additions to the terminal input/output routines, a provision for running FORTRAN jobs detached and a number of other changes to accommodate the new FORTRAN10. Also added were capabilities to assign device names to common directories to facilitate easy access, additions required by the plotter, debugging of buffer synchronization, improvements in the pseudo-interrupt (PSI) handling, and other assorted bug fixes.

EDITORS

A variety of editors are offered so that users will have a choice and to make available any editor that a user may be accustomed to from another PDP10 system. In general, only SOS or TECO for hardcopy terminals and TV for DataMedia display terminals are widely used. We have the TENEX version of TECO and also added the DEC TECO this year. The transition from our local SOS to UTAH-10 SOS is proceeding smoothly and should be completed this month. Error reports and suggestions for new features were solicited from all SOS users and passed along to KKAY@UTAH-10. Similarly, we have made a recent effort to determine the SUMEX community priorities for development of TV; and we are currently conferring with Mr. Pentti Kanerva of IMSSS regarding the future direction of TV development.

BATCH PROCESSING

With the increasing system load, it is advantageous to both the user and the system to perform as much work in non-peak hours as possible. The BATCH facility allows submission of non-interactive jobs to be spooled for later automatic running. BATCH required extensive debugging and modification this year but is now fully operational. Users are being strongly encouraged to consider this option.

MESSAGE PROGRAMS

Bug fixing and streamlining of the SNDMSG (mail sending facility) has continued. We have also adopted a new standard mail reading facility called MSG which is authored/maintained by Mr. John Vittal of the Information Sciences Institute (ISIB). Extensive communication with Mr. Vittal in the developmental stages has lead to a product which serves the SUMEX community needs very well and runs with no problems after local incompatibilities were diagnosed and accommodated in the MSG program.

SEARCH PROGRAMS

A new fast substring search program, XSEARCH, has been written in SAIL by Mr. Scott Daniels of IMSSS/SUMEX. This has been highly successful as a stand-alone search program. It can look through the entire <DOC> directory in about a minute of CPU time. The core code of XSEARCH has also been worked into two library packages: USEARCH which stresses flexible printing of the context of the found string (and is used as the basis for the new HELP program) and PSEARCH which uses TENEX PMAP'ing to increase the search speed even more. A similar very fast algorithm has been incorporated in the WHOIS program by Dr. Lederberg. WHOIS is an often-used program for searching a file containing user names, addresses, and affiliations and the speed-up from the better search algorithm has been a major improvement.

OMNIGRAPH

SUMEX has added a CALCOMP plotter, a Tektronix terminal, and a GT40 light-pen facility to the locally available hardware. The software for the Tektronix was already available in the OMNIGRAPH package from NIH. SUMEX did write a local demonstration program for the cross-hair feature of the Tektronix. The CALCOMP plotter was also available with OMNIGRAPH; but due to differences both in the facilities offered with the particular terminal and in the TOPS10 and TENEX environments, a large scale three-fold programming effort was necessary to get our CALCOMP running. First, a spooler is being written for the plotter (independent of OMNIGRAPH). Second, the PLOTX program of OMNIGRAPH has been debugged and extended for local use: online access is allowed, plot titles are used, x and y may be stretched independently, a clipping routine is added, and the SAIL record data structure is used so that the limit on number of plots is removed. Third, a general CALCOMP control load module was written which currently is used to drive PLOTX but could also be used as a general plotting facility to form the basis of other plotting packages. This module adds more extensions to the plotting capabilities: string prints and line plots with edge checking and arbitrary specification of character codes.

A light pen facility has been added to the OMNIGRAPH code for the GT40 by Mr. Frank Wingate or our staff. This work was done in conjunction with NIH and the original design was followed with all code being incorporated back into the NIH master source files for OMNIGRAPH. A demonstration program was also written for this new feature.

DISPLAY TERMINAL SUPPORT

An increasing number of users have access to Datamedia display terminals. In addition to the TV editor, a variety of programs have been written specifically for these terminals. EDIR places a representation of the user's directory on the screen and allows him/her to point the cursor at a file and give a command letter to indicate desired actions such as Archive, Delete, Undelete, Type (which clears and later refreshes the screen), List, etc. The screen is updated to reflect the current state of each file and a running total of active and deleted file pages is displayed at the top of the screen.

SYSMON displays system loading statistics on the CRT screen and updates the display at regular intervals to reflect changes. included in the display is a ranking of active users according to CPU time used and statistics on i/o use, size of the balance set, idle time, load average, etc.

SCROLL is an editor for creating and storing pictures which can then be reprinted on the screen at any time under program control. The editor facility is especially designed for moving freely among the parts of a picture and is particularly suited for flow-chart drawing. The ability to call pictures which are stored under a given name is very useful in CAI work and other demonstrations where predefined displays can be used as illustrations.

RECORD

The RECORD program written in SAIL by Dr. Robert Smith has become very popular at SUMEX this year and also has been modified to meet the needs of the DENDRAL group for dealing with their GUEST users. RECORD was designed to use the pseudo-teletype facilities for making a file typescript of a terminal session. It has been extended so that the PTY job, once started, can run independently freeing the physical terminal for other work. In the new DENDRAL version, a guest user logging-in on a special directory is automatically interfaced to RECORD to make a typescript of the session (with the user's knowledge). This entire operation is transparent to the user and facilitates later analysis of the session by the DENDRAL staff to learn where the programs need improvement.

PUB

PUB (written in SAIL by Mr. Larry Tesler formerly of SU-AI and currently at XEROX-PARC) is a very powerful documentation preparation language. It is more difficult to use than DEC's RUNOFF but also is more flexible. Last year SUMEX produced a substantially improved set of macros that make use of PUB simpler. This was accompanied by a new manual and a series of well-received PUB introductory classes. New extensions to the PUB macro facility this year include: automatic bibliographic entries from a library with flexible cross-reference printing formats, automatic queuing and placement of tables and figures, multi-column handling, and a marginal notation feature allowing revised or otherwise emphasized text to

be marked. Requests for the SUMEX macro package for PUB have been received from NIH, SRI, AMES, ISI, and IMSSS.

DIABLO

This program is a driver for terminals with a DIABLO or DIABLO-like print mechanism which are used to produce high quality typed output. The DIABLO program is specifically designed for printing PUB output files and it will handle PUB underlining and half-line printing. DIABLO now supports GEN COM as well as DTC terminals. The PUB/DIABLO combination has been upgraded to print a form of proportional spacing which consists of evening the spaces between words when justifying to the right margin. This produced a significant improvement in the appearance of the output. Plans are also on-going to include a hyphenation algorithm in PUB which will be another significant improvement.

TAPE PROGRAMS

Foreign tapes continue to cause time-consuming problems in transferring data to our machine. New tape documentation has been written which explains the tape situation to users and makes recommendations about which of the various tape reading programs to use for various format tapes. Two new tape programs have also been added. MTCOPY from DEC has the ability to handle multiple tapes in a single command. The SUPXEC program by Mr. Ron Roberts of IMSSS was designed to operate like an EXEC for tapes with complete Directory, Copy, and Delete commands available.

A large variety of smaller utilities programs have also been added to the SUMEX catalog. With the file system operating near capacity, one emphasis this year has been on file management programs. Utilities are now available: to plot the age of a user's files, to allow deletion/archiving of all files older than a cut-off date, to find all files newer than n hours, to clean up wasted file space in text files, to find files with multiple versions, to rename files, to copy selected portions of files together, to view the actual character codes in a file, to check the file-descriptor-block (FDB) for a file, to find all new public files, to find the exact location of a file, to recognize partial filenames, to print files in reverse order, to provide a handle on files too large to edit conveniently, to encrypt text files, to convert the character case of files, etc.

Another area of utilities development has been programs to manage personal calendars or to serve as on-line reminder systems. And, of course, an area of continuing development is informational utilities. The WHOIS program to give name/address/phone number and project affiliation information on users may well be the most-often used of these utility programs.

II.A.2.h USER ASSISTANCE AND CONSULTING

User consulting continues to play a key role at SUMEX. Because of the geographic distribution of our users where they may have little or no direct contact with computer experts and the nature of the user community in which many non-computer professionals are involved, the user consulting load is higher than on most similar systems. While direct individual consultation has been a major component of the effort and will continue to be, other solutions to the problem are continually sought. A number of approaches have proven useful:

- 1) Foster interactions among users themselves to help each other learn and to solve particular problems. This has been the case with the new statistical packages. The staff is available to fix program bugs but in fact has little expertise in the use of the packages. So the groups expressing interest have been put in touch with each other. This effort has been quite successful.
- 2) Among the systems coupled by networks (ARPANET currently), focus maintenance responsibility for particular pieces of software in the groups which developed them or where an extensive expertise already exists. Typically the author/maintainer is best able to deal with user problems. With the ARPANET to provide communication access at a non-local site, the duplicated effort in a local staff trying to familiarize themselves with imported programs for bug fixing and consulting is minimized. An example of this is the shift this year from our local SOS editor program which had been developed by a former staff member and was no longer actively supported by any current staff member to a version of SOS which is well-maintained at UTAH-10 by Kevin Kay. He analyzed our version, incorporated the improved features into his version, and assumed support of it at SUMEX. This is an increasingly common phenomenon among the TENEX sites on the ARPANET. Other software maintained by one site for all the sites includes MSG, REDUCE, SAIL, INTERLISP and PUB. In general, one local staff member is the primary contact for each of these and does handle routine problems and questions but does so in close communication with the author. And in fact, for both SOS and SAIL, users are encouraged to go directly to the authors; SOS has a built-in GRIPE facility which sends the user message to SOS@SRI.
- 3) Employ media which reach a large number of users rather than dealing on an individual basis. This includes writing of documentation (see Section II.A.2.j on page 40) and holding classes and tutorials. Last year, several classes were given (PUB, SAIL, and machine language). These were successful; and we have followed up with an advanced SAIL course, a BAIL lecture, and a planned repeat of the PUB class. More informal meetings have also been held to discuss issues such as choice of a programming language and efficient programming in languages such as SAIL and INTERLISP. Participants in last year's Workshop requested an INTERLISP tutorial which Dr. R. Carhart of the DENDRAL project gave. A number of tutorials on languages and other aspects of SUMEX use are planned for the June 1976 Workshop.
- 4) Use other techniques for interactive on-line teaching. Interactive

computers have been used in a number of areas for computer-assisted-instruction (CAI) applications. We may be able to adapt some of these techniques and will be studying possibilities of a training mode for selected programs (possibly a text editor or the EXEC). In this we are fortunate to have close ties with IMSSS, a well-known leader in the field of CAI.

II.A.2.i INTRA-COMMUNITY COMMUNICATION

Help System:

A substantial problem for users not intimately familiar with the system (and even those who are) is how to locate the appropriate documentation on-line to answer a particular question as it arises. Many times a staff member or other user is not available to help so we have been developing various forms of on-line assistance or "help". After considering a number of help schemes, early this year we put up a temporary help program which optionally printed a general information file and then interactively helped the user search through the names of the <DOC> files with a keyword search. This approach is rather effective at SUMEX where long filenames composed of as many keywords as possible are used to identify the information files. The interim help program was well-liked by users. A new more sophisticated version is currently being tested.

It offers all the facilities of the simple version but it also has extensions in three basic directions. First, in addition to checking the filenames for the given keyword, it also checks the contents of several general information files, e.g., the file listing all programs with a one-line summary of each, a file containing an on-line index of the jsys's, a file containing entries on new programs, and a file containing the TENEX command equivalents for TOPS10 commands. If the keyword is found in the keyword line of any of these entries then the individual entry is printed out. These files were all existing documentation files which needed only slight format changes to be used in this way. Second, the user can specify certain standard keywords which allows him to do more specialized searching of the relevant data base for the topic. Third, when the filenames of the <DOC> directory are searched for the keyword and a list of possibly relevant files has been produced then the user has several options. The contents of any of the files can be searched for further keywords or the entire files (or selected pages) can be typed on the terminal, put together in a new file, or listed on the printer. This allows the user to browse around and tailor-make a new document with just the desired pieces found.

Bulletin Board System:

Another kind of user communication system has been under design and implementation for some time at SUMEX. Some information is of a more informal or transient nature than that comprising files suitable for the <DOC> directory. Other types of information have relevance for only a subset of the community such as intra-project communication about program design, external users, etc. We have maintained a <BULLETINS> directory for system-related "bulletins" (see page 228 for a current directory listing) but have needed a more general facility serving public and private bulletin boards as well as allowing users to selectively direct their interests to particular subsets of the information. Such a bulletin-board system would fill the gap between the <DOC> directory, intended for permanent documentation, and the mail system, where each user (and the system) has his own mailbox. It would complement the help system, providing quick access to such intermediate information.

An on-line "bulletin board" system is nearly ready for release to the SUMEX community. The following is a brief description of the system in its preliminary implementation. We expect it to grow and change as users begin to use it and identify additional communication needs.

The system has a number of bulletin boards available. A public bulletin board encompasses system-wide information. It is for community use and has bulletins on new features of system programs, announcements, corrections, progress reports, suggestions, queries, and comments of general interest to the SUMEX community. There may be need for other public bulletin boards such as for the AIM workshop depending on the overall volume of information involved. Private bulletin boards may focus on individual projects (e.g., DENDRAL, ONET, MOLGEN, etc.), subsystems, or other subgroupings of the community.

Bulletins are filed on the bulletin boards under topics, which may have any number of subtopics. They each have an expiration date, because some information may be of a temporary nature. In general, the kind of bulletin posted is what used to be sent out in multiple copies with a SNDMSG distribution list to a number of users, and would now be posted on the bulletin board pertaining to that interest group or project.

Two programs operate on the bulletin-board files. POST is the equivalent of SNDMSG or ADDMSG with extra editing and display features and doubles as a bulletin poster. POST can send copies of the same bulletin to a user list and bulletin boards at the same time. When not used to post a bulletin, it behaves like SNDMSG. The program BBD performs other inquiry and editing functions on bulletin boards. It is designed to behave as the TENEX EXEC does, for consistency with existing command conventions and ease of use. There are also some similarities between BBD and message-reading programs like MSG and BANANARD. A BBD user will be able to connect to the bulletin board of his choice and:

Get a directory listing of topics and bulletins

Type a (set of) bulletin(s) by number or topic name

Ask to see the first 5 lines only of each

Copy bulletins to other files (including TTY: or LPT:)
in message format

In each of the above, the user can narrow his range of bulletins by asking BBD only for those that meet certain criteria, such as:

New bulletins

Bulletins filed under topics on his "interest list"

Bulletins written by a particular author or group of authors

Deleted bulletins

Expired bulletins

Bulletins posted before or after a specified date

Bulletins with a desired phrase in the message or subject line

or combinations of the above.

There will be a notification system whereby bulletin-board users are notified once per day of new bulletin arrivals. The user tells BBD to add a given topic to his "interest list". He is notified of new arrivals only for those topics, unless his interest list is "all topics". There is a separate interest list for each bulletin board. The notification system can easily be extended to be an automatic reminder system. One could post a bulletin on a "Reminders" bulletin-board. The bulletin would be sent to those to whom it was addressed when it expired.

Bulletin-board users will likely want BBD to assume a few things for them. They may specify a bulletin-board to connect to upon entering the program (default is the main bulletin-board), or they may never want to see anything but what is on their interest list. BBD will have a "Defaults" command which will set these things up on a per user basis.

Each bulletin board will have a manager, who has special privileges. He will be able to create and destroy topics, delete and undelete bulletins, reorder the bulletins in each topic, change the assignment of bulletins to topics, and change expire dates. Everybody will be able to delete, undelete, refile, and change the expire dates of bulletins they author.

A first version of the bulletin board system is in final checkout. We expect to release it to the community this summer and to continue development based on user reactions and needs during the upcoming year.

II.A.2.j DOCUMENTATION AND EDUCATION

SUMEX has set and maintained high standards for documentation. This year we achieved virtually complete documentation of all available programs. The list of the <DOC> directory reported last year contained 142 files. That has now increased to 220 files (see Appendix F on page 218); many of which have been updated and reorganized. All of the general information files have also been updated during the course of the year. SUMEX is probably the best documented PDP10 site on the ARPANET. However, we have long recognized the fact that the usefulness of the documentation is severely limited by the ease of access to the particular bit of information that a user is currently seeking. As the volume of documentation increases, more information is available in an abstract sense but may perversely become less available in real terms because of the difficulty of finding it. The HELP and BULLETIN BOARD systems are designed to help overcome these problems.

This year, SUMEX submitted a 25-page entry to the ARPANET Resource Handbook (September 1975) which contains a variety of information on the ARPANET host systems. Our entry includes a description of the SUMEX facility and projects, a list of the software available for export with a policy statement on the procedure for obtaining this software, and a summary of the major areas of interest at SUMEX. A copy of this material is attached as Appendix G on page 230.

As a follow-up to last year's very successful SAIL introductory classes, this year a series of more advanced SAIL and Machine Language seminars was given by Dr. R. Smith. These covered the interface of SAIL to the PDP10 host machine and timesharing system, SAIL program debugging, and implementation and efficiency considerations. Mr. John Reiser of SU-AI gave a seminar on the features and use of the new SAIL debugging system, BAIL. A SAIL Tutorial has been written by Dr. Nancy Smith of SUMEX which will be published shortly by SU-AI along with a reprinting of the standard SAIL document plus the TENEX-SAIL Manual and the new BAIL Manual. Finally, Dr. Nancy Smith, in conjunction with the improved macro facilities and documentation, gave an introductory class in using PUB.

II.A.2.k SOFTWARE COMPATIBILITY AND SHARING

Over the past year, in our commitment to software importation where possible rather than reinvention, we have encountered numerous experiences in the sharing of software. At SUMEX many avenues exist for sharing between the system staff, various user projects, other facilities, and vendors. In the past without communication networks, the system vendor served as the focal point for distribution of most software to user sites. Since the process of distributing tapes (and particularly of handling bug reports and user suggestions) was very slow, it was common for sites to take a version of a program and then modify and maintain it locally. This caused a proliferation of home-grown versions of software. Similar impediments have existed to the dissemination of user software. User organizations like SHARE and DECUS have helped to overcome these problems but communication is still cumbersome. The advent of fast and convenient communication facilities coupling communities of computer facilities has

the potential of making a major difference in facilitating inter-group cooperation and to lower these barriers.

Recently, the TENEX sites on the ARPANET have been interacting increasingly with each other to develop new software systems. This functions effectively to build communication around the network and promote a functional division of labor and expertise. 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. Certain common problems are now regularly discussed on a multi-site level. We continue to draw significant amounts of system software from other ARPANET sites reciprocating with our own local developments.

Currently the number of sites involved is relatively small and the interactions are informal. It may be that this informality is an essential ingredient to making this process work, much as friendships among people develop. It may be the bureaucracy of vendor systems and procedures (which do have useful fallout in uniform documentation, interfaces, etc.) which caused the proliferation of home-grown systems in the past. Indeed our own attempt at building a SAIL library may have foundered because we tried to be too formal about it.

We began an effort last year to accumulate useful SAIL library routines from the various groups which have been working with this language (Stanford AI, IMSSS, SRI, NIH, USC-ISI, etc.). It has been somewhat surprising that so little communication of SAIL library programs has taken place - it is almost literally true that each user has his own stock of tools in private procedure libraries. We sent a letter to interested groups soliciting inputs on a basis which attempted to balance the problem of assuring library quality and integrity against establishing so high a threshold for quality and polish that individuals are not motivated to cooperate. Despite the willingness of active community to share time and ideas on an individual basis, there have been virtually no external entries to the library in response to our efforts.

In other areas, however, where individuals have undertaken the design of major software components, mutual design cooperation between sites has a growing list of examples of success. Undoubtedly the particular personalities involved play some role as well as the orientation of the funding agencies.

Certainly the TENEX operating system itself is an example of community cooperation although there has been some tendency for localization because of BB&N's rigidity. Other examples of cooperation mentioned earlier include SOS, MSG, PUB, the batch system, and our substantial efforts to contribute to software exportability through developing the MAINSAIL system. This latter effort has received very enthusiastic support from many quarters of the computer community. Other noteworthy examples encountered this past year include the following.

When Mr. John Reiser began writing the BAIL debugger for SAIL, he

was contacted and agreed to design the program for maximum compatibility between the SU-AI system, TENEX, and standard TOPS10. This effort was quite successful. BAIL was written in such a way as to use each of the operating systems optimally with no compromise in program design. One estimate of the extra development time involved was only 10 per cent. The important ingredients are complete program comments, modular design, and no unnecessary system dependent code. It can be contrasted with other programs written at SU-AI. For example, in the process of designing a bulletins system, SUMEX learned of the AP NEWS Service program at SU-AI which has many features similar to our design for bulletins. The program was studied and proved to be very difficult to transfer and adapt because of the choice of language and the degree of dependence on the SU-AI home-brew operating system. Both BAIL and NEWS were written at SU-AI and both are well-written programs by other criteria.

MLAB and OMNIGRAPH -- Even without the facilities of the ARPANET and with all the compatibility problems of TOPS10/TENEX program sharing, our interactions with the NIH Division of Computing Research and Technology concerning MLAB and OMNIGRAPH have been mutually beneficial. SUMEX has sent code for a "TENEX" conditional compilation switch to NIH which has been incorporated into the source files. Also, the light pen and plotting development work done here this past year has had close communication with NIH.

With very careful organization by Dr. R. Smith, the export of TENEX-SAIL has proceeded with very good community cooperation. A special directory called <XSAIL> has been created (which can be accessed by the ANONYMOUS feature of the FTP file transfer program so that there is no need for exchanging passwords). This directory contains ALL the necessary files for exporting the SAIL system. All sites running TENEX-SAIL were individually contacted and requested to appoint a local contact for communications purposes.

The openness of the ARPANET communication facilities encourages some members to copy pieces of software without the author's knowledge thereby defeating the necessary more orderly processes of maintenance and upgrade and in some cases losing proper attribution for the program's development. This has occurred with a number of the programs that we have available for export. It is difficult to control such behavior without at the same time limiting access for cooperating members of the community. We try to discourage it by pointing out the self-defeating effects.

Another continuing effort is in the maintenance of software compatibility with DEC. The PA1050 program (for TOPS10 emulation) is an important part of the software for each TENEX site. SUMEX made a search for all local versions of PA1050 and combined the best features. Much new work was also done. This version has been made available to other TENEX sites - IMSSS and SRI are running it with direct cooperation and other sites have copied it without informing us. Since almost all sites using TENEX are doing government-funded work and this is an obligatory condition for ARPANET access, we have not felt it necessary until now to take strenuous (and possible costly) measures to protect this software. We will, however, review this problem periodically.

A new aspect of DEC compatibility arises with the announcement of the TOPS-20 operating system which has been developed by DEC for their KL-20 machine. The current 2040 system is a relatively small system but will be followed by larger members of the 20-family. TOPS-20 is based on TENEX. It appears that ARPA may be transferring support from BBN to DEC for system development and the ARPANET interface for the system. This has the potential for greatly decreasing the compatibility problems since both TOPS-10 and TOPS-20 will be under DEC control. On the other hand, DEC has implemented a variety of minor changes already (new JSYS's, different file name notation, etc.) which are causing a divergence between TOPS-20 and TENEX that may well lead to greater compatibility problems than exist now. We noted these possibilities in the decision to remain with TENEX and implement the dual processor system. The timing of DEC's evolution of TOPS-20 with larger scale processors is uncertain as is the rate with which the ARPANET community might move in that direction. There are many existing KA-10 and KI-10 machines running TENEX for which there are no current prospects of replacement. Over the next few years we feel our decision was correct, especially in view of budgetary constraints. However, we are sensitive to remaining as parallel as possible with the mainstream of the community and will actively pursue this goal.

II.A.3 RESOURCE MANAGEMENT

Over the past year, the SUMEX project has devoted a substantial part of its effort toward its community-building role in recruiting new projects, promoting interactions between user projects, and encouraging dissemination of running performance programs to medical scientists. The following summarizes specific aspects of SUMEX-AIM community management activities.

II.A.3.a MANAGEMENT COMMITTEES

The SUMEX-AIM resource is constituted to attempt to bring into closer contact collaborating health research groups from around the country. This mission entails both the recruitment of appropriate research projects interested in medical AI applications and the catalysis of interactions among these groups and the broader medical community. As this effort is not a unilateral undertaking by its very nature, we have created several management committees to assist in administering the various portions of the SUMEX resource. As defined in the SUMEX-AIM management plan adopted at the time the resource grant was awarded, the available facility capacity is allocated 40% to Stanford Medical School projects, 40% to national projects, and 20% to system development and related functions.

Within the Stanford aliquot, Dr. Lederberg has established an advisory committee to assist him in selecting and allocating resources among projects appropriate to the SUMEX mission. The current membership of this committee is listed in Appendix H.

For the national community, two committees serve complementary functions. An Executive Committee oversees the operations of the resource as related to national users and makes the final decisions on authorizing admission for projects. It also establishes policies for resource allocation and approves plans for resource development and augmentation within the national portion of SUMEX. The Executive Committee oversees the planning and implementation of the AIM Workshop series and assures coordination with other AIM activities as well. The workshops are being carried out under Dr. S. Amarel of the Rutgers Computers in Biomedicine resource. The current membership of the Executive committee is listed in Appendix H.

Under the Executive Committee functions an Advisory Group representing contact with medical and computer science research relevant to AIM goals. The Advisory Group serves several functions in advising the Executive Committee; 1) recruiting appropriate medical/computer science projects, 2) reviewing and recommending priorities for allocation of resource capacity to specific projects based on scientific quality and medical relevance, and 3) recommending policies and development goals for the resource. The current Advisory Group membership is given in Appendix H.

These committees are actively functioning in support of the resource. Meetings to date have been held by telephone conference for the most part owing to the size of the groups and to save the time and expense of personal travel to meet face to face. These "missings" (a term coined by Dr. Licklider), in conjunction with terminal access to related text materials, have served quite well in accomplishing the agenda business and facilitate greatly the arrangement of meetings. A few technical problems occasionally attend such sessions such as poor telephone reception for some members but in general this approach is quite satisfactory. The key to success seems to be a) fairly short and not too infrequent sessions, b) a firm agenda, c) mail distribution of relevant documents, d) computer network backup for exchange of information, and e) informality and personal rapport of the members. Other solicitations of advice requiring review of sizable written proposals are done by the mails.

II.A.3.b NEW PROJECT RECRUITING

As a result of the public announcements of the SUMEX resource, NIH contacts with prospective grantees, and personal contacts by the staff or committee members, a number of additional projects have been admitted to SUMEX; others are working tentatively as pilot projects or are under review. We have prepared a variety of materials for the new user ranging from general information such as is contained in the brochure (Appendix I) to more detailed information and guidelines for determining whether a user project is appropriate for the SUMEX-AIM resource. Dr. E. Levinthal has prepared a questionnaire to assist users seriously considering applying for access to SUMEX-AIM (see Appendix J). Pilot project categories have been established both within the Stanford and national aliquots of the facility capacity to assist and encourage projects just formulating possible AIM proposals pending a formal review.

The projects newly admitted over the past year include (see Section IV for more detailed descriptions):

National -

- 1) Chemical Synthesis Project (SECS); Dr. T. Wipke (University of California at Santa Cruz)
- 2) Language Acquisition Modelling (ACT); Dr. J. Anderson (University of Michigan)

As an additional aid to new projects or collaborators with existing projects, we have a limited amount of funds which are being used to support terminals and communications needs of users without access to such equipment. We are currently leasing 6 terminals and 4 modems for users as well as 4 foreign exchange lines to better couple the Rutgers project into the TYMNET and a leased line between Stanford and U. C. Santa Cruz for the Chemical Synthesis project.

II.A.3.c STANFORD COMMUNITY BUILDING

During the past year, the Stanford community has undertaken several efforts to encourage interactions and sharing between the projects centered here. Beginning in the fall term, Professor Feigenbaum organized a seminar class with the goal of assembling a handbook of AI concepts, techniques, and current state-of-the-art. This project has had enthusiastic support from the students and substantial progress made in preparing many sections of the handbook. An outline of the material to be prepared along with an indication of the status of each article can be found in Appendix B on page 180. Several examples of completed articles are given in Appendix A on page 166.

A second community-building effort was a "mini AI conference" held at Stanford in January 1976. This 3 day series of meetings featured presentations by each of the local projects and comparative discussions of approaches to current problems in AI research such as knowledge representations, production system strategies and rule formation, etc. A brief summary of the conference is attached as Appendix C on page 194.

II.A.3.d RESOURCE ALLOCATION POLICIES

As the SUMEX facility has become increasingly loaded, a number of diverse and conflicting demands have arisen which require controlled allocation of critical facility resources (file space and central processor time). We have already spelled out a policy for file space management; an allocation of file storage is defined for each authorized project in conjunction with the management committees. This allocation is divided among project members in any way desired by the individual principal investigators. System allocation enforcement is implemented by project each week. As the weekly file dump is done, if the aggregate space in use by a project is over its allocation, files are archived from user directories over allocation until the project is within its allocation.

As described under TENEX monitor software development, we have been using a primitive CPU scheduling algorithm intended to ensure that no one user gets more than a fair share of the machine when other users are contending. With the implementation of TENEX 1.33 this summer, the pie-slice allocation system will be available to more rigorously ensure CPU allocation by project and community allocations.

As also mentioned earlier, we have categorized users in terms of access privileges. These comprise fully authorized users, pilot projects, guests, and network visitors in descending order of system capabilities. We want to encourage bona fide medical and health research people to experiment with the various programs available with a minimum of red tape while not allowing unauthenticated users to bypass the advisory group screening procedures by coming on as guests. So far we believe we have

had little or no exploitation compared to what other sites have experienced, perhaps on account of the personal attention that senior staff gives to the logon records. However, the experience of most other computer managers behooves us to be cautious about being as wide-open as might be preferred for informal service to pilot efforts and demonstrations. We will continue developing this mechanism in conjunction with management committee policy decisions.

II.A.3.e AIM WORKSHOP SUPPORT

The Rutgers Computers in Biomedicine resource (under Dr. Saul Amarel) is actively working on plans for the second AIM workshop this June. The current plans call for a four day series of meetings covering a range of topics related to artificial intelligence research, medical needs, and resource sharing policies within NIH. The SUMEX facility will act as a prime computing base for the workshop demonstrations. We hope to have the new dual processor system in operation for the meetings. A final decision will depend on progress over the next week in completing the debugging of the initial system and our ability to assure reliable operation. We are in the process of working with Rutgers to provide backup modes for program demonstrations in the event of computer system problems.

II.A.4 FUTURE PLANS

System Development:

In the next year much work remains to complete the dual processor system. We must complete installation, evaluate its performance in terms of increased throughput, identify and fix excessive waiting for monitor interlocks, and optimize system scheduling and resource handling. We plan to implement a mutual interrupt facility between the two machines and to implement a bus switch allowing I/O devices to be moved easily between the two machines. This will increase our ability to keep the system running in the case of a processor failure by reconfiguring to a single processor mode.

We plan to continue evaluation of system hardware bottlenecks and to pursue avenues to eliminate them. We know that disk space is currently a problem and are trying to augment the system through user project cooperation. Other limiting resources over the next year may be memory and swapping space.

We will install version 1.33/1.34 of TENEX with necessary dual processor and KI-10 modifications in order to stay current with other TENEX sites and to improve resource allocation controls among the AIM community members.

We plan to improve the batch processing capability for those jobs which need not run interactively. A current system has helped to move system loading from prime time to off hours. We plan to extend facilities for error handling and more flexible job scheduling.

We will continue to refine the Executive program and capabilities for guest users in conjunction with the TENEX 1.33 upgrade.

We will also investigate ways of improving network communication services. This will include attempts to optimize our current facilities for users through better ties to the networks and selective lines to tie individual users into more advantageous access points. We will also continue to explore other network and communication alternatives as they become available over the next year. Specific goals include improved response times and increased output speeds. TYMNET will be starting 1200 baud service soon and we would like to make it possible for users to take advantage of the higher output speed.

MAINSAIL:

We are awaiting the funding of the MAINSAIL project to allow initial export of this language system. We have established contacts with numerous outside groups interested in this machine-independent language ranging from university research projects to industry. (The university's research projects office is handling any problems or opportunities that may arise from proprietary values of these products, in accordance with established procedures). We have proposed an initial list of target

machines including PDP-10, PDP-11, and Nova. We plan to develop an exportable, documented form of the system for each of these environments and to test them in conjunction with appropriate collaborating user sites.

Adaptive User Interfaces:

We plan to continue work toward a more adaptive system for users including both simplifying access for non-expert users and anticipating default parameter conventions of individual users. We are now in the process of defining system calls which will make user specifications accessible to utility programs in a uniform way.

Software Facilities and Libraries:

There is a continuing need for improved documentation and self-learning facilities for various aspects of the system and of available programs. We will be up-grading this material, particularly as it relates to the inexperienced user.

We will continue to up-grade the various DEC-originated subsystems to the newest versions to increase the chance of compatibility. We have recently done this with FORTRAN and MACRO and will bring the other programs along as soon as possible. The whole issue of compatibility is one which will receive continued attention. We will also increase our mutual ties in software sharing with the TENEX and AI communities.

Some requests to look into additional software subsystems have been received and we will consider mounting them if the community develops a definite need.

Informal Information Access:

One characteristic of the SUMEX community is the diversity of information, formal and informal, which flows around the system or is available from users. We will continue to work on the HELP and BULLETIN BOARD systems to capture that information and direct it to other interested individuals. We will be working on capabilities both to ease the entry and cataloging of information and to assist in guiding the user to that subset which is of interest to him at a given time. These user-oriented lookup protocols are, of course, strongly related to the problems of adaptive user interfaces to the system and each will benefit from the experience of the other.

Community Management:

We will continue to work with the management committees to recruit the additional high quality projects which can be accommodated and to evolve resource allocation policies which appropriately reflect assigned priorities and project needs. We hope to make more generally available information about the various projects both inside and outside of the

community and thereby to promote the kinds of exchanges exemplified earlier and made possible by network facilities. The AIM workshops provide much useful information about the strengths and weaknesses of the performance programs both in terms of criticisms from other AI projects and in terms of the needs of practicing medical people. We plan to use this experience to guide the community building aspects of SUMEX-AIM.

II.B SUMMARY OF RESOURCE USAGE

The following data give an overview of the resource usage from May 1975 through April 1976. There are three sub-sections containing data respectively for 1) resource usage by community (AIM, Stanford, and system), 2) resource usage by project, and 3) Network usage data.

II.B.1 RELATIVE SYSTEM LOADING BY COMMUNITY

The SUMEX resource is divided, for administrative purposes, into 3 major communities: user projects based at the Stanford Medical School, user projects based outside of Stanford (national AIM projects), and systems development efforts. As defined in the resource management plan approved by BRP at the start of the project, the available resource will be divided between these communities as follows:

CPU Usage - Stanford	40%
AIM	40%
Staff	20%
File Space - Stanford	27,000 pages(*)
AIM	27,000 pages
Staff	13,500 pages

(*) One TENEX page is 512 36-bit words or 2560 text characters)

An additional allocation of approximately 30,000 pages serves system files including documentation, subsystems, monitor, etc.

The monthly usage of CPU and file space resources for each of these three communities relative to their respective aliquots is in the plots in Figure 8 and Figure 9. Our diurnal variations in loading have retained the same characteristics as previously, with a bimodal distribution reflecting the complementary loads from the east coast and the west coast.

Figure 8.
CPU USE BY COMMUNITY

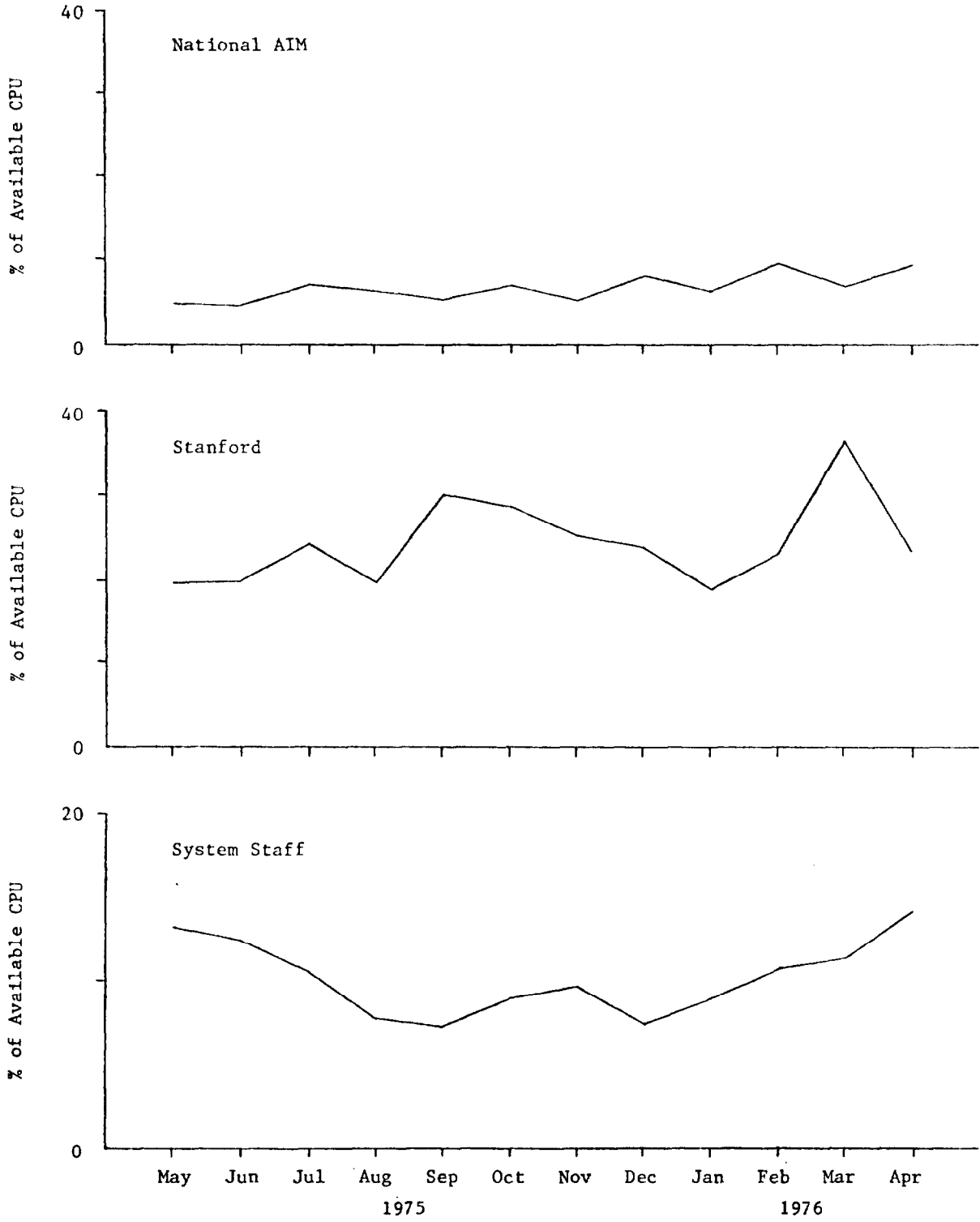
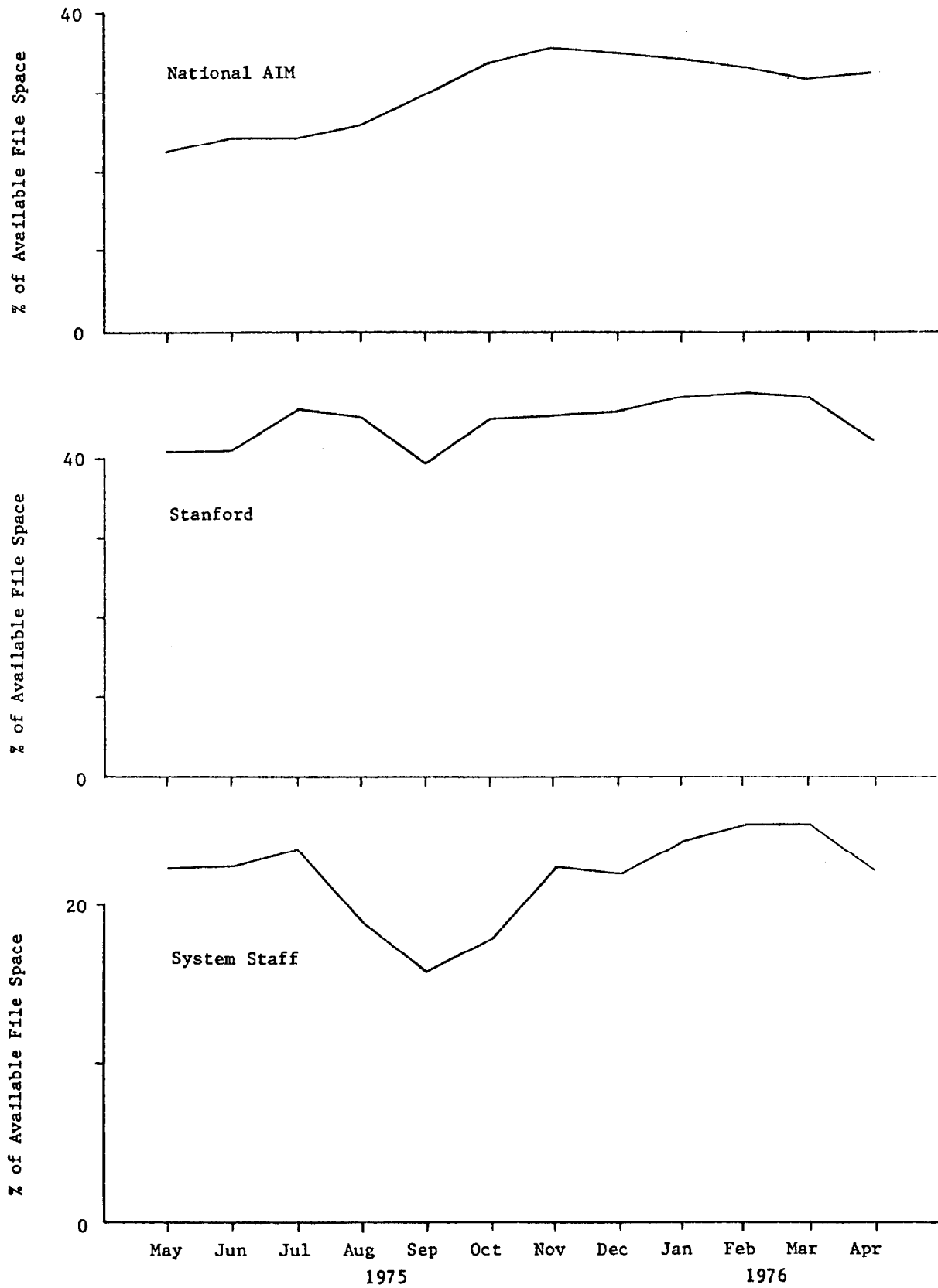


Figure 9.
FILE SPACE USE BY COMMUNITY



II.B.2 INDIVIDUAL PROJECT AND COMMUNITY USAGE

The table following shows average resource usage by project in the past grant year. The data displayed include a description of the operational funding sources (outside of SUMEX-supplied computing resources) for currently active projects, average monthly CPU consumption by project (Hours/month), average monthly terminal connect time by project (Hours/month), and average file space in use by project (Pages/month, 1 page = 512 computer words). Averages were computed for each project for the months between May 1975 and April 1976.

RESOURCE USE BY INDIVIDUAL PROJECT

STANFORD COMMUNITY	CPU (Hrs/mo)	CONNECT (Hrs/mo)	FILE SPACE (Pages/mo)
1) DENDRAL PROJECT "Resource Related Research Computers and Chemistry" NIH RR-00612 (3 yr award) \$240,967 this year	68.41	1574	18280
2) MYCIN PROJECT "Computer-based Consult. in Clin. Therapeutics" HEW HSO-1544 (3 yr award) \$163,965 this year	20.76	494	5959
3) PROTEIN STRUCT MODELING "Heuristic Comp. Applied to Prot. Crystallog." NSF DCR74-23461 (2 yrs.) \$88,436 total	19.45	296	2452
4) PILOT PROJECTS (see reports in Sec IV.B.1)	14.12	433	3459
	-----	-----	-----
COMMUNITY TOTALS	122.74	2797	30150

NATIONAL AIM COMMUNITY

1) SECS PROJECT "Chemical Synthesis" NIH proposal pending	10.32	196	3284
2) INTERNIST PROJECT (DIALOG) "Computer Model of Diagnostic Logic" HEW MB-00144 (3 yrs.) \$167,168 last year	7.64	209	4705
3) Higher Mental Functions "Computer Models in Psychiatry and Psychother." NIH MH-27132 (2 yrs.) \$67,000 this year	1.94	85	1299

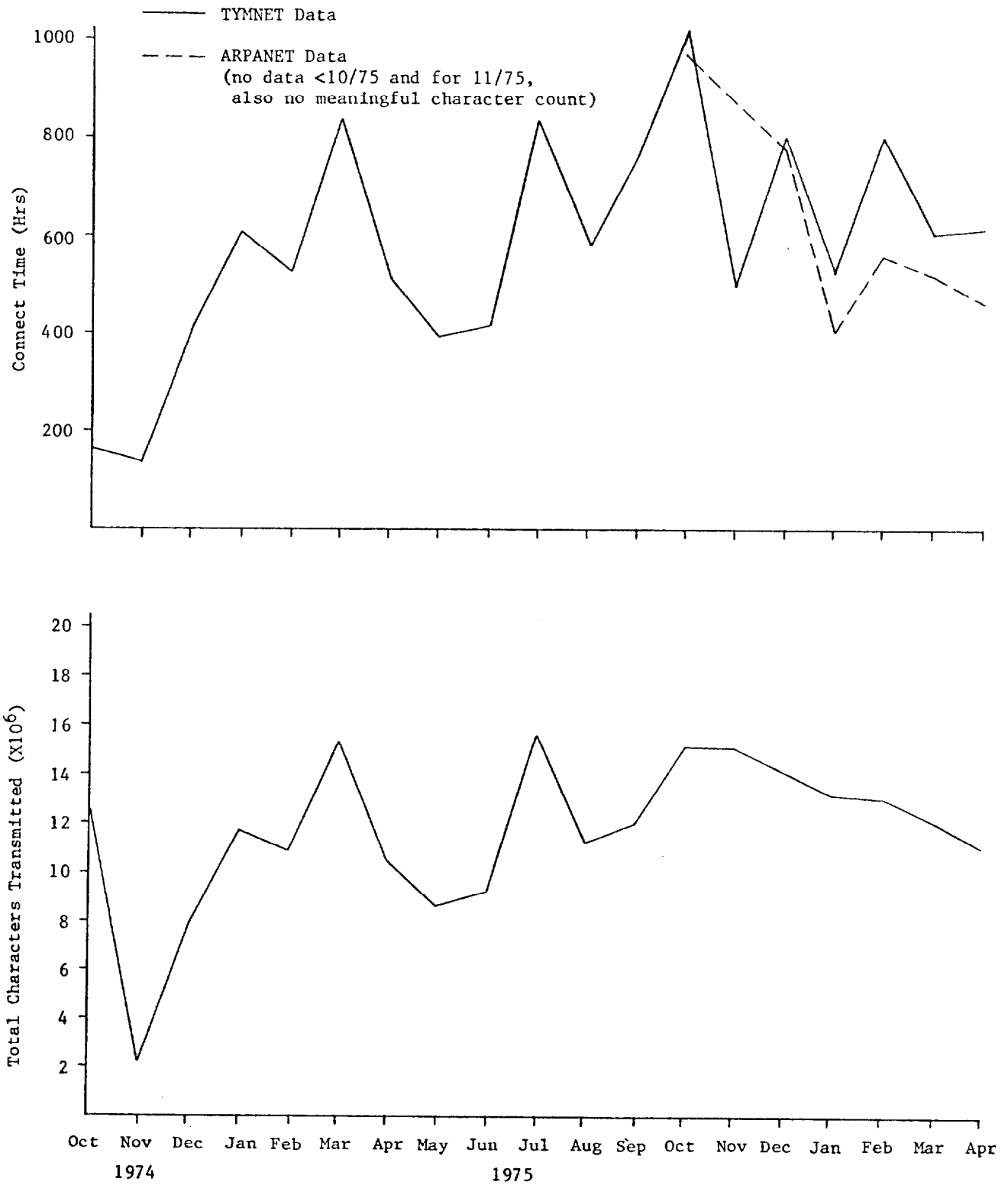
4)	ACT PROJECT "Language Acquisition Modeling" NIMH \$20,000 this year	2.77	55	559
5)	MISL PROJECT "Medical Information Systems Laboratory" HEW MB-00114 (3 yrs.) \$248,793 this year	0.98	45	773
6)	RUTGERS PROJECT "Computers in Biomedicine" NIH RR-00643 (3 yrs.) \$314,880 this year	12.17	446	8174
7)	AIM PILOT PROJECTS	0.27	9	66
8)	AIM Administration	1.88	76	2897
		-----	-----	-----
	COMMUNITY TOTALS	37.97	1121	21757
SUMEX STAFF AND SYSTEM				
1)	Staff	50.99	2126	14453
2)	System & Operations	63.71	3688	27033
		-----	-----	-----
	COMMUNITY TOTALS	114.70	5814	41486
		=====	=====	=====
	RESOURCE TOTALS	275.41	9732	93393

II.B.3 NETWORK USAGE STATISTICS

NETWORK USAGE PLOTS

The plots in Figure 10 show the major billing components for SUMEX-AIM TYMNET usage. These include the total connect time for terminals coming into SUMEX and the total number of characters transmitted over the net. The ratio of characters received at SUMEX to characters sent to the terminal is about 1:(10-14) over the past couple of months. Also shown for recent months is a plot of ARPANET connect time which tracks the corresponding data for TYMNET usage fairly closely. No data for "Character" transmission is available for ARPANET since file transfers and terminal traffic use different byte sizes and these data are not resolved and maintained for the ARPANET.

Figure 10.
SUMEX-AIM NETWORK USAGE



II.C RESOURCE EQUIPMENT SUMMARY

The following table gives a list of the items of equipment purchased to date for the SUMEX resource along with details on vendor, description, price, and date.

ITEM	QUANTITY	DESCRIPTION	MANUFACTURER	MODEL NUMBER	DATE INSTALLED	DATE ACCEPTED	PURCHASE PRICE	SOURCE FUNDS
KI-10 CPU	1	Central processor, including console	Digital Equipment Corporation	KI-10	3/1/74	4/24/74	\$178,500	NIH
	1	Central processor, including console	Digital Equipment Corporation	KI-10	4/15/76	5/7/76	\$203,138	NIH
Memory	3	Core memory (64K words including 4 MC-10 memory ports)	Digital Equipment Corporation	MF-10G	3/1/74	4/24/74	\$224,910	NIH
	1	Core memory (64K words including 4 MC-10 memory ports)	Digital Equipment Corporation	MF-10G	11/74	12/74	\$ 63,484	NIH
	1	Memory port multiplexer	Digital Equipment Corporation	MX-10	8/74	9/74	\$ 4,770	NIH
Clock	1	Programmable clock	Digital Equipment Corporation	DK-10	3/1/74	4/24/74	\$ 2,678	NIH
	1	Programmable clock	Digital Equipment Corporation	DK-10	4/15/76	5/7/76	(incl. in second processor)	
Disk System	1	Single double density disk controller	Digital Equipment Corporation	RP-10C	3/1/74	5/1/74	↓	↓
	1	Memory data channel	Digital Equipment Corporation	DF-10	3/1/74	4/24/74		
	4	Double density disk drives and disk packs	Digital Equipment Corporation	RP-03	3/1/74	4/24/74	\$108,153	NIH
	3	Double density disk drives and disk packs	Digital Equipment Corporation	RP-03R	2/75	3/75	\$ 44,636	NIH

ITEM	QUANTITY	DESCRIPTION	MANUFACTURER	MODEL NUMBER	DATE INSTALLED	DATE ACCEPTED	PURCHASE PRICE	SOURCE FUNDS
Swapping Storage	2	Fixed head disk with 1.7M word capacity and 4 track parallel access	Digital Development Corporation	A-7312-D-8	1/75	3/75	\$ 37,206	NIH
	1	Special systems controller for DDC disks	Digital Equipment Corporation	RES-10	10/74	11/74	\$ 81,090	NIH
DEC Tapes (TU-56)	1	DEC tape control	Digital Equipment Corporation	TD-10	3/1/74	4/24/74		
	1	Dual DEC tape drive	Digital Equipment Corporation	TU-56	3/1/74	4/24/74	\$ 17,850	NIH
Magnetic Tapes (2 x TU-30)	1	Magnetic tape controller	Digital Equipment Corporation	TM-10A	3/1/74	4/24/74		
	2	Tape transports	Digital Equipment Corporation	TU-30	3/1/74	4/24/74	\$ 31,238	NIH
Line Printer	1	Special systems line printer control for Data Products 2410	Digital Equipment Corporation	Special	6/74	7/74	\$ 7,208	NIH
	1	Line printer with 96 character drum, vertical format control, parity check	Data Products	2410	6/74	7/74	\$ 18,963	NIH

ITEM	QUANTITY	DESCRIPTION	MANUFACTURER	MODEL NUMBER	DATE INSTALLED	DATE ACCEPTED	PURCHASE PRICE	SOURCE FUNDS
GT-40	1	Graphics terminal	Digital Equipment Corporation	GT-40	3/1/74	4/24/74	\$ 11,156	NIH
Line Scanner	1	Data line scanner	Digital Equipment Corporation	DC-10A	3/1/74	4/24/74		
	1	8-line unit	Digital Equipment Corporation	DC-10B	3/1/74	4/24/74	\$ 16,275	NIH
TYMNET Interface	1	PDP-10 TYMNET communications controller	TYMSHARE		8/74	10/74	\$ 50,774	NIH
ARPANET Interface	1	BB&N ARPANET/KI-10 interface and VDH	Bolt, Beranek & Newman		1/75	2/75	\$ 21,200	NIH
PDP-11/10	1	Communications processor	Digital Equipment Corporation	PDP-11/10	2/75	3/75	\$ 13,445	NIH

ITEM	QUANTITY	DESCRIPTION	MANUFACTURER	MODEL NUMBER	DATE INSTALLED	DATE ACCEPTED	PURCHASE PRICE	SOURCE FUNDS
Terminals	1	Terminal	Data Terminals Communications	DTC-300	3/18/74	6/74	\$ 4,597	NIH
	2	Terminals - Execuport portable with carry case	Computer Transceiver Systems, Inc.	311-3	3/18/74	6/74	\$ 6,402	NIH
	6	Terminals - elite CRT with edit capabilities	Datamedia	2500	9-10/74	11/74	↓	↓
	2	Terminals - elite CRT with edit capabilities	Datamedia	2500	8/75	8/75		
	2	Terminals - elite CRT with edit capabilities	Datamedia	2500	12/10/75	1/29/76		
Keyboards	3	Keyboards, special, for leased Datamedia elite 2500 CRT terminals at - NIH Rutgers Univ. Washington Univ.	Datamedia	70DVK7019	11/74	12/74	\$ 1,118	NIH

ITEM	QUANTITY	DESCRIPTION	MANUFACTURER	MODEL NUMBER	DATE INSTALLED	DATE ACCEPTED	PURCHASE PRICE	SOURCE FUNDS
Modems	16	Auto answer modems	Prentice Electronics	P-113B	5/6/74	5/6/74	↓	↓
	5	Auto answer modems	Prentice Electronics	P-1200/150	5/6/74	5/6/74		
	3	Auto answer modems	Prentice Electronics	P-1200/150	4/2/76	4/2/76		
	5	Originate modems	Prentice Electronics	P-1200/150	5/6/74	5/6/74		
	4	Modem enclosure with loopback switch and cables	Prentice Electronics	P-100	5/6/74	5/6/74		
	4	Modem enclosures for 8 modems with cables, power supply, digital loopback, line loopback, indicator lights	Prentice Electronics	P-850	5/6/74	5/6/74		
	3	Acoustic coupler modems	Prentice Electronics	DC-22	3/74	3/74		
	3	Modem enclosure with line loopback switch to house P-103F modems	Prentice Electronics	P-100	3/74	3/74		
Oscilloscope	1	Oscilloscope	Tektronix, Inc.	475DM43	1/75	1/75	\$ 3,476	NIH

II.D PUBLICATIONS

Publications for the SUMEX staff have included papers describing the SUMEX-AIM resource and on-going research:

- [1] Carhart, R.E., Johnson, S.M., Smith, D.H., Buchanan, B.G., Dromey, R.G., and Lederberg, J, Networking and a Collaborative Research Community: a Case Study Using the DENDRAL Programs, ACS Symposium Series, Number 19, COMPUTER NETWORKING AND CHEMISTRY, Peter Lykos (Editor), 1975.
- [2] Levinthal, E.C., Carhart, R.E., Johnson, S.M., and Lederberg, J., When Computers Talk to Computers, Industrial Research, November 1975

Mr. Clark Wilcox was asked to chair the session on Languages for Portability at the DECUS DECsystem10 Spring '76 Symposium. A description of his work will appear in the proceedings.

In addition as reported earlier, a substantial effort has gone into developing, upgrading, and extending documentation about the SUMEX-AIM resource, the SUMEX-TENEX system, and the many subsystems available to users. These efforts include a number of major documents (such as SOS, PUB, and TENEX-SAIL manuals) as well as a much larger number of document upgrades, user information and introductory notes, an ARPANET Resource Handbook entry (see Appendix G), and policy guidelines (see Appendix F, Appendix I, and Appendix J). Publications for individual user projects are summarized in the respective reports (see Section IV).

III RESOURCE FINANCES

III.A REFERENCE TO BUDGETARY DETAILS

The budgetary materials for the SUMEX project covering past actual costs, current performance, and estimates for the next grant year are submitted in a separate document to the NIH.

III.B RESOURCE FUNDING

The SUMEX-AIM resource is essentially wholly funded by the Biotechnology Resources Program [*]. The various collaborator projects which use SUMEX are independently funded with respect to their manpower and operating expenses. They obtain from SUMEX, without charge, access to the computing and, in most cases, communications facilities in exchange for their participation in the scientific and community building goals of SUMEX.

[*] Except for the participation by Stanford University in accordance with general cost-sharing, and for assistance to SUMEX by other projects with overlapping aims and interests.

IV RESOURCE PROJECT DESCRIPTIONS

The following are inputs from the various user projects currently in the SUMEX-AIM community. These project descriptions and comments are the result of a solicitation for contributions sent to each of the project Principal Investigators requesting the following information:

- I) Summary of research program
 - A) Technical goals
 - B) Medical relevance and collaboration
 - C) Progress and accomplishments
 - D) Current list of project publications
 - E) Funding status (current funding level and pending applications or renewals)

- II) Interactions with the SUMEX-AIM Resource
 - A) Examples of collaborations and medical use of programs through networks
 - B) Useful contacts and cross fertilization with other SUMEX-AIM projects (via workshop, messages, terminal links, etc.)
 - C) Critique of resource services

The text which follows on the various projects is primarily the responsibility of the indicated project leaders.

IV.A FORMALLY APPROVED PROJECTS

IV.A.1 STANFORD USERS

IV.A.1.a DENDRAL PROJECT

DENDRAL PROJECT

Principal Investigators: Profs. C. Djerassi (Chemistry),
J. Lederberg (Genetics), and E. Feigenbaum (Comp. Sci.)

(Grant NIH RR-00612-06, 3 years, \$240,967 this year)

OVERVIEW

In the period August, 1975 to July, 1976 the DENDRAL programs and the gas chromatography/mass spectrometry (GC/MS) data system have made significant progress toward the goals stated in the research proposal. This report of progress is organized in three parts, corresponding to the three specific aims of our December, 1973, proposal: (PART 1) Enhancing the power of the mass spectrometry resource, (PART 2) Developing performance and theory formation programs, and (PART 3) Applying the computer programs and instrumentation to biomedically relevant structure elucidation problems.

The DENDRAL project, one of the major users at Stanford of the SUMEX-AIM computer facility, has also been forming its own community of remote users. This national "EXODENDRAL" community has already provided valuable contributions to program development and both the community and contributions are expected to grow at an increased rate.

PART 1: ENHANCING THE POWER OF THE M.S. RESOURCE

1.1 Introduction

Our grant proposal requested funds for significant upgrading of our capabilities in mass spectrometry. The goals of this upgrading were to provide routine high resolution mass spectrometry (HRMS), combined gas chromatography/low resolution mass spectrometry (GC/LRMS) and to develop a combined gas chromatography/high resolution mass spectrometry (GC/HRMS) facility. In addition, this would provide the capability for new experiments in the detection and utilization of data on metastable ions. These capabilities would then be available as required for application to our wider goal, solution of biomedical structure elucidation problems of a community of researchers.

The upgrading included several items of hardware and software development, as follows: 1) Acquire stand-alone computer support for the mass spectrometer because existing facilities were inadequate and very expensive; 2) convert existing software, written in the PL/ACME language

into FORTRAN so that it would run on the new system; 3) develop new software as required for the demanding task of GC/HRMS; 4) provide hardware and software for semi-automatic acquisition of data on metastable ions. The initial development phase of this upgrading included performance tests to determine the capabilities and limitations of the GC/HRMS system to define the scope of problems to which it can be applied. The past year's efforts (year two of the DENDRAL grant) have culminated in accomplishment of many of the above goals for development. In the first year, the computer system (a Digital Equipment Corp. PDP 11/45) was purchased, installed and is now operating routinely in conjunction with the mass spectrometer (a Varian-MAT 711) and an auxiliary PDP 11/20 system. Program conversion and modification for the initial version of the software system was completed and the computer system now provides complete stand-alone support for our experiments in mass spectrometry. Over the past year we have developed further our philosophy of data acquisition and reduction based on computed models of the actual performance of the mass spectrometer. This was and is necessary for routine automated collection and reduction of combined GC/HRMS data with minimal operator intervention in the procedures.

The system development is motivated by two goals. First, the system must be robust in the sense that it continue to operate under a variety of changing conditions, including intermittent misbehavior of the mass spectrometer. This ensures that the system can recover from hardware or software error conditions to prevent fatal "crashes" of the system and resulting loss of data. Second, the system must automate the GC/HRMS task. The volume of data acquired in GC/HRMS experiments can be efficiently handled only when every spectrum can be acquired and reduced for final output by the system without manual intervention. We are successful in these goals because we have written the software to determine the actual performance of the mass spectrometer and to have subsequent calculations based on that measured performance, as opposed to some hypothetical ideal.

We are now providing GC/HRMS service on a limited basis as we improve the system. The time devoted to system development and testing will slowly diminish over the next year, leaving additional time for analysis of mixtures obtained in our own work and that of our collaborators. We have deferred implementation of the metastable system (see below) while the GC/HRMS development is continuing, although we have completed the hardware and much of the software for the system.

Because we view GC/HRMS as the most important new capability of our mass spectrometer/computer work, the requirements of GC/HRMS have guided development of the software system. These requirements include continuous automatic monitoring of instrument performance to avoid wasting time collecting poor or erroneous data. By approaching GC/HRMS with an electrical recording system, we can monitor the instrument continuously, both during initial setup and during the course of the GC/HRMS experiment. While photographic recording may capture more of the signal, it is vulnerable to fluctuations in sample and instrument behavior in addition to the difficulties in reading the data from film for computer analysis. Major sections of the software and how they interact among one another are summarized below.

During the past year the routine production usage of the HRMS data has become a reality. The direct utilization of the system for the acquisition of high resolution mass spectrometry data typically occupies 6 hours per day. This figure does not include time for the post-processing of data, retrieval of data from the archival data base, or for the generation of duplicate print outs of selected data. These demands add 1 to 2 hours of system service each day to the total high resolution system requirements.

Low resolution mass spectral data whether it be smoothed from high resolution data or obtained directly as low resolution data, places additional time demands upon the data system. High to low resolution conversion, low resolution plotting, and low resolution spectral library searching have all generated a need for increasing amounts of system time.

In an effort to utilize the data system more completely during non-prime time, batch and spooling mechanisms have been constructed. The high resolution spectral reviewing mechanism may be actuated and then left unattended while the hard-copies are being generated. The high to low resolution conversion process contains a mechanism for the generation of a low resolution plotting spool which can be played without operator intervention. Batch procedures have been written which provide for the archival of newly acquired spectral data in the archival data base.

As with any system as large as the high resolution system there is a continual need for system maintenance and minor software upgrades. A wider range of data acquisition and analysis places new demands upon the system which require further modification of the software.

The net result of the production demands has been to reduce the amount of system time available for the development of new software facilities. Software development and production compete for the available system time reducing the productivity of both the chemical user and the software developer. This competition can be drastically reduced if software development can proceed on a machine separate from that on which production is done. The SUMEX PDP-10 and TENEX operating system provide a more tractable medium for development than does the restricted environment of the PDP-11.

A major factor in the ease with which programs can be constructed is the ease with which text can be manipulated. The TV-EDIT program which is available on the PDP-10 has proven to be effective for this task. This program provides an extremely flexible text editing system for display terminals. The mechanics of program construction can be greatly simplified by the utilization of this facility. Typically all major (more than a few changes) text modification of programs are carried out on the PDP-10 using TV-EDIT and then transferred to the PDP-11. Thus even the task of writing FORTRAN programs is simplified even though there exist FORTRAN incompatibilities between the two machines.

While TV-EDIT has reduced development demands on the PDP-11 by eliminating PDP-11 text editing sessions, the problem of program compilation and debugging remain. Clark Wilcox, of the SUMEX staff, has provided an effective solution to this problem with the development of the

MAINSAIL (Machine Independent SAIL) compiler. This compiler provides the user with a powerful machine independent structured language. Not only is the compiler machine independent, but exhibits superior execution speeds and storage requirements as compared to the DOS 9 FORTRAN which has been used previously.

The combination of TV-EDIT and MAINSAIL has proven to be an effective method for the development of software for the PDP-11s within the PDP-10 environment. Most debugging can be carried out on the PDP-10 and then transferred to the PDP-11s for final debugging of machine-dependent facilities. The class of machine-dependent facilities includes device drivers and interaction with the operating system. The class of machine-independent facilities includes analysis algorithms, file manipulation, and most other programs which need development. This means that the amount of time required on the PDP-11 for program development can be reduced significantly using the aforementioned process, leaving more time for production demands.

1.2 Summary

As the above hardware and software improvements are being made we will continue evaluation of the GC/HRMS system in parallel with its actual application to real problems. GC/HRMS is a relatively new and difficult technique for routine application. In order to use it effectively, we will have to exert some effort toward determining and optimizing the performance of the many elements of the system, the GC, the MS, and the computer hardware and software.

PART 2: DEVELOPING PERFORMANCE AND THEORY FORMATION PROGRAMS TO ASSIST IN BIOMEDICAL STRUCTURE ELUCIDATION PROBLEMS

2.1 Introduction

The Heuristic DENDRAL computer programs assist with structure elucidation problems by helping interpret mass spectra and helping generate structures that are consistent with data obtained from a variety of spectroscopic and physical/chemical courses. The Meta-DENDRAL programs assist with rule formation problems in cases where the rules of mass spectrometry are not known.

Both the interpretation and rule formation programs are written as interactive tools to be controlled by professionals to combine the professional's judgment with the computer's combinatorial power.

2.2 CONGEN.

The CONGEN[48,53] program represents a significant extension of a program which has developed over the last several years, the cyclic structure generator[40,41]. The purpose of CONGEN is to assist the chemist in determining the chemical structure of an unknown compound by 1)

allowing him to specify certain types of structural information about the compound which he has determined from any source (e.g., spectroscopy, chemical degradation, method of isolation, etc.) and 2) generating an exhaustive and non-redundant list of structures that are consistent with the information. The generation is a stepwise process, and the program allows interaction at every stage; based upon partial results the chemist may be reminded of additional information which he can specify, thus limiting further the number of final structures.

CONGEN fits with the other DENDRAL programs as a "backstop" solution to structure elucidation problems. If the mass spectrum of an unknown compound is available, then CLEANUP and MOLION could be used, but if the general class of the compound is not known, PLANNER has no starting point from which to work. In such cases, structural information can be extracted manually from the spectrum and given to CONGEN for analysis. Because CONGEN makes no assumptions about the source of this information, other spectroscopic or chemical techniques may be used to supply supplemental data.

At the heart of CONGEN are two algorithms whose accuracy has been mathematically proven and whose computer implementation has been well tested. The structure generation algorithm[31,37,40,41] is designed to determine all topologically unique ways of assembling a given set of atoms, each with an associated valence, into molecular structures. The atoms may be chemical atoms with standard chemical valences, or they may be names representing molecular fragments ("superatoms") of any desired complexity, where the valence corresponds to the total number of bonding sites available within the superatom. Because the structure generation algorithm can produce only structures in which the superatoms appear as single atoms (we refer to these as intermediate structures), a second procedure, the imbedding algorithm[48,53] is needed to expand the superatoms to their full chemical identities.

These two routines give the chemist the ability to construct structures from a given set of molecular "building blocks" which may be atoms or larger fragments. By itself, this capacity is of limited utility because the number of final structures can be overwhelming in many cases. Usually, the chemist has additional information (if only some general rules about chemical stability, which the program has no concept of) that can be used to limit the number of structural possibilities. For example, he may know that because of a compound's stability, it cannot contain a peroxide linkage (O-O) and thus the programs need not consider such structures when there are two or more oxygens in the "building block" list.

In the past year CONGEN has reached the level of a practical production program which can aid chemists, both locally and at remote network sites, in solving the structures of drug-related compounds and natural products. The development of this program during the year has been strongly guided by the difficulties and new requirements which have appeared as it was applied to a wide variety of cases, and its efficiency and usefulness have increased dramatically. We report here the details of the modifications and additions we have made to CONGEN, and the effects they have had on its utility. Also, because of the rich repertoire of

structure modification and testing functions available within CONGEN, we have found it to be an invaluable "laboratory" for the testing of new ideas, and we briefly describe several pilot projects which form the basis for future research. Discussion of applications of CONGEN to problems of biochemical interest is included in Part 3.

NEW CAPABILITIES FOR THE USER. There have been several additions to CONGEN which are visible to the user and which generally increase the flexibility and power of the program. These include:

- 1) Making CONGEN aware of aromaticity, a chemical property of molecules which results from certain combinations of double bonds in rings. Aromaticity has a profound effect upon both the chemical reactivity and symmetry properties of molecules, and CONGEN can now be directed to detect aromaticity in its output structures, to compensate for the difference between the actual symmetry of an aromatic system and the symmetry which appears in the graph representing it, and to distinguish aromatic from non-aromatic atoms when it tests GOODLIST and BADLIST entries.
- 2) Giving the user the ability to type "?" to any prompt in the program, which results in a summary of the possible inputs. In some cases this summary is a list of possible commands, while in others it is a short explanatory message. A new interactive teletype-input routine was developed which makes it easy to include such help messages in the program, and which mimics the handy command-recognition and command-completion features of the TENEX operation system.
- 3) Including new specifications in the EDITSTRUC language for describing substructural features. The user can now declare a bond in a substructure to be an "anybond", which means that the atoms at the termini are connected but that the multiplicity of the connection is unspecified. This is especially handy when defining substructures containing aromatic portions because bond multiplicity is an indistinct concept in aromatic systems. Another new structural element which can be specified is a "linknode", a node which stands for a variable-length chain of atoms of the given type rather than a single atom. The minimum and maximum lengths of such a chain can be specified as well. The linknode feature is useful for defining constraints on ring fusions and other constraints such as Bredt's rule which depend on path length. Other extensions have been made internal to CONGEN which will shortly be reflected in the user-level language of EDITSTRUC. These include numerical inequalities involving node properties (e.g., "the number of H's on atom 3 is greater than the number of H's on atom 5") or linknode lengths (e.g., "the sum of the lengths of linknodes 2 and 6 is greater than 5"), and greater control over the number of fittings found for a GOODLIST constraint (e.g., the ability to distinguish between "the number of N's in six-membered rings" and "the number of six-membered rings containing N").
- 4) Allowing greater flexibility in the selection of terminal type. This choice controls the output of structural drawings so they are best suited to the user's terminal. Several different types of character-oriented and graphics-display terminals are now supported.

- 5) Making CONGEN accessible from the GUEST login account at SUMEX. This involved preventing a GUEST user from reaching certain critical points in CONGEN which would allow greater system access than is normally authorized for guests. We can now offer trial access to CONGEN via the guest mechanism without worrying about SUMEX misuse.
- 6) Creating a BATCH command for CONGEN. This allows the user to submit time-consuming, compute-bound calculations to the batch-processing facility of SUMEX. The computation is then run automatically at off-hours when it will not overload the system resources. The user can now run CONGEN in its interactive mode to input all of his data and then submit the large tasks to BATCH for overnite processing.
- 7) Including a pruning function MSPRUNE which is used to test a list of candidate structures for consistency with a set of observed peaks from a mass spectrum. The candidates are typically generated by CONGEN using structural data from other sources. The user specifies the observed MS peaks (high- or low-resolution, or a combination of both) along with a set of constraints on the allowed cleavage processes. MSPRUNE retains only those candidates which can account for the observations via one of these allowed processes. The constraints speak of the number of bonds broken and the number of steps in a process, the proximity of pairs of cleaved bonds (i.e., whether or not two adjacent bonds can break in a given process), the multiplicity or aromaticity of each cleaved bond and the possible neutral transfers. MSPRUNE is the first CONGEN function which can aid directly in the interpretation of "raw" spectral data.

2.3 Meta-dendral Rule Formation Programs

The INTSUM program [34] is in routine, production use to assist in interpretation of the mass spectra of new classes of molecules (see Part 3 for details). When the mass spectrometry rules for a given class of compounds are not known, the INTSUM, RULEGEN and RULEMOD programs can help a chemist formulate those rules. Essentially, these programs categorize the plausible fragmentations for a class of compounds by looking at the mass spectra of several molecules in the class. All molecules are assumed to belong to one class whose skeletal structure must be specified. Also, the mass spectra and the structures of all the molecules must be given to the program.

INTSUM collects evidence for all possible fragmentations (within user-specified constraints) and summarizes the results. For example, a user may be interested in all fragmentations involving one or two bonds, but not three; aromatic rings may be known to be unfragmented; and the user may be interested only in fragmentations resulting in an ion containing a heteroatom. Under these constraints, the program correlates all peaks in the mass spectra with all possible fragmentations. The summary of results shows the number of molecules in whose spectra there is evidence for each particular fragmentation, along with the total (and average) ion current associated with the fragmentation.

The RULEGEN program attempts to explain the regularities found by

INTSUM in terms of the underlying structural features around the bonds in question that seem to "drive" the fragmentations. For example, INTSUM will notice significant fragmentation of the two different bonds alpha to the carbonyl group in aliphatic ketones. It is left to RULEGEN to discover that these are both instances of the same fundamental alpha-cleavage process that can be predicted any time a bond is alpha to a carbonyl group.

The RULEMOD program modifies and condenses the set of rules produced by INTSUM and RULEGEN together. It looks at the negative evidence associated with each candidate rule in order to select the best ones, then merges rules that seem to explain the same breaks (if possible). The program was substantially improved in several ways, as described in the next section.

2.3.1 INTSUM Improvements

Transfers of arbitrary neutral species can now be specified as part of the mass spectrometry processes, instead of transfers of hydrogen atoms alone. This capability increases the utility of the program in at least two ways: first, it allows a chemist to control the program better -- to produce the kinds of results that are more chemically meaningful -- and second, it allows the program to explore more complex processes within its space and time limitations. For example, carbon monoxide and water were listed as plausible neutral molecules to transfer in or out of fragments for the triketoandrostanes. Thus, the processes are listed with and without these transfers, just as chemists prefer, instead of showing loss of CO as a set of two breaks around the keto group, or loss of H₂O as loss of oxygen (breaking the C=O bond) accompanied by loss of two hydrogens. What is more, the program can now produce these results without violating its chemical heuristics of (a) not breaking adjacent bonds, and (b) not breaking double bonds. This economy also pays off in increasing the complexity of the processes that can be considered. Because loss of CO, for example, is a result of a transfer instead of the result of breaking two bonds, the number of bonds broken in accompanying processes can be increased by two.

Another INTSUM improvement was to increase the options for initial data filtering. Thresholding is too simple for many problems, so we now provide an option to cluster peaks and select the n largest peaks from each cluster.

The format of the input data is also now less strict than before. We have written programs to read spectra in Aldermaston format. And we have merged CONGEN's Editstruc package into the INTSUM setup routines to allow a chemist to associate structures with spectra interactivity. This greatly decreases the chances of error in setting up the input data.

Several modifications were also made to the program to increase its efficiency, e.g., processing all intensities as integers (between 0 and 1000).

2.3.2 RULEGEN Improvements

The evaluation of prospective rules in RULEGEN guides the entire rule generation procedure. To tune this procedure, we modified the evaluation function in several ways and compared the resulting sets of rules. We were looking for an objective way of telling the program to keep rules general, but "not too general". The current evaluation function is substantially improved as a result.

Because the RULEGEN program searches such a large space of partial and complete rules, it requires large amounts of computer time (sometimes more than 60 cpu minutes). Thus, we have investigated several improvements for efficiency alone. In addition, we have made the program easier to set up and run in batch mode to reduce the chemist's personal time investment. And we have made the program easily restarted from any intermediate point -- to protect the chemist from machine failures.

2.3.3 RULEMOD Improvements

At the time of the last annual report RULEMOD was a new program still in its experimental stages. Since then we have added new subprograms and integrated the program with other programs to make it a useful and necessary part of Meta-DENDRAL.

Two new subprograms greatly improve RULEMOD's performance. (1) A program to add specifications to rules was completed. It looks for plausible ways of making a rule more specific in order to decrease the number of counterexamples to the rule. (2) A complementary program to make rules more general was also completed. The program tries to find ways to reduce the number of descriptors on nodes of subgraphs in order to increase the breadth of applicability of rules. Its major constraint is that it cannot make any change that would increase the number of counterexamples. Both of these subprograms make the final rules much closer to rules that chemists approve of.

The subprogram that merges rules was also improved. The program tries to merge pairs of rules into a more general form for economy and clarity of rules. Its major constraint is that no explanations are lost, i.e., all the data points explained by the initial pair of rules will still be explained after merging. Formerly we insisted that the more general form must cover all the same data points as the initial rules, but this was found to be too narrow a constraint. By giving the program a more global view of the entire set of rules, we can let the more general, merged form explain fewer data points as its component rules as long as other rules explain the remainder.

PART 3: APPLICATIONS TO BIOMEDICAL STRUCTURE ELUCIDATION PROBLEMS

3.1 Introduction

In our grant proposal we discussed the application of the instrumentation and computer programs described above to the study of molecular structure problems in a variety of biomedical applications areas. This is our primary research area, and we discussed specific

classes of problems and compounds for investigation. We also made it quite clear that our facilities would be made available to wider community of collaborators/users as our resources permitted. Both categories of application, i.e., within our own group, and with an outside group, are described in some detail below.

Our last annual report described several steps taken to encourage a broad community of researchers to use our facilities. For example, we sent a questionnaire to members of the American Society for Mass Spectrometry, Committee III on Computer Applications, and a follow-up letter to persons indicating a desire to know more about access to our programs. The same note has been sent to several other persons whom we know from personal contacts might be interested. Because of the nature of their investigations, many of these people receive NIH support. Several of our publications (e.g., [45]-[49]) mention the availability of our programs. In addition, through individual contacts and formal presentations at conferences we have been encouraging outside use of the programs.

The availability of SUMEX as a mechanism for resource sharing has made it possible for us to extend access to our programs to a number of people. Without SUMEX, this access would be impossible, and most of our programs (those which are not easily exportable) could be used only by ourselves.

3.2 Applications by Professor Djerassi's Research Group

Our existing grants, outlined below, mesh well with our instrumentation and program development under the present award. Under NIH Grant GM06840 we have been studying natural products from marine sources with major emphasis on terpenoids and sterols. For this work we have been dependent on the use of our 711 instrument for high resolution mass spectrometry which we require for the identification of all new compounds, many of which are present in only very small quantities. We are particularly anxious to have access to GC coupled with a high resolution mass spectrometer because we hope to be able to screen large numbers of marine animals for their sterol content using this technique.

We are currently engaged in intensive efforts in analysis of mixtures of marine sterols involving our computer-based procedures. The program for the development of the computer operated and assisted system of marine sterol structure analysis has been planned to proceed in three stages:

- 1) Analysis of all literature published concerning marine sterols so that a complete listing of known sterol structures and organisms studied could be compiled.
- 2) Collection, evaluation, digitization and computer file construction for the mass spectra of all known marine sterols, followed by the institution of a computer operated file search sequence for direct analysis of marine sterol GC-MS data.

- 3) The application of the INTSUM, RULEGEN, and RULEMOD programs to the computer file of marine sterol spectra so that a series of fragmentation rules can be extracted for use in the generation of possible structures from mass spectral data for new marine sterols, that is, sterols whose mass spectra cannot be matched with any spectra contained in the computer search file.

3.3 Applications of Programs by External Scientists

The DENDRAL project, still one of the major users of the SUMEX-AIM computer facility, has formed a small community of regular, remote users. This "exodendral" community has continued to provide valuable contributions to program development, although the growth of this community has had to be slowed in response to increasing demands by other projects upon the SUMEX-AIM facility. As an example, for the months of September 1975 to February 1976, the number of CPU hours used by exodendral persons amounted to at least 8 percent of the CPU hours used by the DENDRAL project. There are currently four remote chemist-users whose groups regularly use CONGEN in their day to day work. Additionally, there are several remote users who use their accounts on an occasional basis, or who access SUMEX-AIM via the GUEST mechanism.

The SUMEX-AIM facility has grown markedly in number of projects over the past year. Due to this increase in system loading; the DENDRAL project, which had previously been able to offer trial usage of its programs to almost any chemist who expressed a need to use the programs, has found itself in the unfortunate position of having to carefully screen potential collaborators. Those chemists who have been granted access, have been requested to restrict their usage to off-prime time hours. CONGEN, the DENDRAL program which receives most of this usage, has evolved in a manner designed to try to remedy the system loading problem which can be created by the enthusiasm of its chemist-users. Since a typical, long GENERATE, PRUNE or IMBED within CONGEN can be very time consuming, as well as a voracious consumer of CPU cycles, a provision to permit a user to easily take advantage of SUMEX-AIM's off-hour batch processing has been implemented. A CONGEN user can now interactively set up his problem, and when ready to commence with a time consuming procedure, can, from within CONGEN, request automatic submission to BATCH, to be run late at night. The CONGEN users' also benefit from this ability, in that they no longer must leave a terminal tied up during the sometimes hour-long compute times. This development then, can be viewed as responding to CONGEN users' needs as well as being an effort by the DENDRAL project to be conscientious in its resource-sharing responsibilities.

Following is a brief summary of the major users of CONGEN over the past year, as well as notes on chemists who contacted us about trial usage of the programs.

Dr. Clair Cheer, Professor of Chemistry, University of Rhode Island, Kingston, Rhode Island. Dr. Cheer is on sabbatical leave from the University of Rhode Island to the Stanford University Chemistry Department. He has, in recent work with Professor Djerassi's group,

demonstrated the utility of CONGEN in the identification of (+)-Palustrol, a tricyclic sesquiterpene alcohol from the marine Xeniid Cespitularia viridis (Cheer, Djerassi et. al., Tetrahedron, in press). Dr. Cheer plans to continue his work with CONGEN once he returns to Rhode Island in December.

Dr. Jon Clardy, Professor of Chemistry, Iowa State University. Dr. Clardy read of CONGEN in an article appearing in the Journal of the American Chemical Society and contacted Professor Djerassi concerning the possibility of using the program from Iowa. He was offered GUEST access during the winter of 1975, but has not yet had an opportunity to evaluate the potentials of the program.

Dr. Douglas Dorman, Eli Lilly Corp., Indianapolis, Indiana. Dr. Dorman's research involves the identification and characterization of drug related compounds by chemical and spectrographic methods. Using primarily the NMR and C13 NMR spectra of these various compounds, Dr. Dorman has found CONGEN to be a time-saving adjunct to his structure elucidation work.

Dr. H.M. Fales, National Heart and Lung Institute, Bethesda, Maryland. Dr. Fales, along with Doctors Sanford Markey and Peter Roller had a joint account set up for them in April of 1975. Most of the use of this account came during late summer at which time Dr. Fales experimented with the use of CONGEN for assistance in the elucidation of the structure of a novel quinolinone, known to be tumorigenic. Although the crystal structure had been solved at the time of his usage of CONGEN, Dr. Fales felt that the program produced an abundance of useful ideas. The main problem initially faced by Dr. Fales in using CONGEN was in getting a feel for problem size and the effects of various constraint types.

Professor Kenneth Gash, California State College at Dominguez Hills. Professor Gash is a professor of chemistry who is on temporary leave to Small College, the research branch of Dominguez Hills. Dr. Gash did some of the original work, in 1965, with Professor Morton Munk, on the structure elucidation program developed at Arizona State University. Dr. Gash has been reviewing some of the problems originally done with Munk's program and has been studying input, output and constraint capabilities found in CONGEN. He has generally concluded that CONGEN provides an excellent tool for the chemist to use in structure elucidation problems subject to the constraint of slow system response time.

Mr. Neil A. B. Gray, King's College, Cambridge, England. Mr. Gray, following a three week visit to the Stanford chemistry department, requested copies of all the current DENDRAL programs to be sent to him in England. He is a chemist who has been working in areas related to developments in various of the DENDRAL programs, and hopes to be able to benefit from work already done at Stanford. His current interest in intelligent constraint application during structure elucidation merges well with one of the directions in which CONGEN is tending to develop. Unfortunately, Mr. Gray does not have access to an ARPANET or TYMNET node to access SUMEX-AIM directly. Therefore, all collaboration has had to be carried on by mail.

Dr. Jerrold Karliner, Ciba Geigy Corporation, Ardsley, New York. Dr. Karliner and his research group at Ciba-Geigy have become regular users of CONGEN in their day-to-day operation of a research laboratory. Dr. Karliner is a completely self-taught user of CONGEN, and has served to encourage others to request permission to use this program.

Dr. Milton Levenberg, Abbott Laboratories, Chicago, Illinois. Dr. Levenberg has been an occasional user of CONGEN as an adjunct to his work as head of a mass-spectrometry laboratory. Primary usage has been to provide assurance that the proposal of a structure for a compound on the basis of chemical and spectroscopic evidence has not overlooked other plausible possibilities.

Dr. Gino Marco, Ciba Geigy Corporation, Greensboro, North Carolina. Dr. Marco heard about CONGEN during a company seminar presented by Dr. Karliner. After a brief trial use via the GUEST mechanism, Dr. Marco requested an account for use by his group of metabolic and organic chemists. Dr. Marco's research group studies unknown insect metabolites by micro-IR and micro-NMR methods, and attempts structure elucidation based on these forms of spectroscopic analysis. Testing the utility of the program before implementing it for day to day use, Dr. Marco discovered that CONGEN could greatly narrow the alternatives of complex metabolic conjugates which had to be considered in a typical elucidation problem. They have established a leased line to the nearest TYMNET node, and expect increased CONGEN usage in the future.

Professor G. Minole, Italy. Professor Minole has been active in elucidation of structures of marine natural products, an area of interest which overlaps with our own. We have provided, by written communication due to absence of network access, sets of structural alternatives in current problems being studied by Professor Minole. We have used some of the mass spectrometric prediction functions of our DENDRAL programs to determine which structures in a set of possibilities could yield the observed mass spectral data.

Professor Nogi Nakanishi, Department of Chemistry, Columbia University. Professor Nakanishi is one of the most active and productive persons engaged in structure elucidation activities. He has developed an active interest in CONGEN and is collaborating with us on several novel problems. One of these problems has involved the structure of the active component of defense secretions of an insect (termite). Other defense secretion components are under investigation as we explore structural alternatives based on current data.

Dr. David Pensak, DuPont de Nemours and Company, Wilmington, Delaware. Indirectly requested information about CONGEN through a letter written by his immediate superior to Professor Lederberg. Dr. Pensak has been offered GUEST access, and has just begun a potential collaboration with a DENDRAL group which is studying model builders and their production of reliable geometries for certain types of molecules.

Professor Manfred Wolff, University of California at San Francisco. Dr. Wolff is chairman of the Department of Pharmacological Chemistry, and inquired as to the possibilities of accessing SUMEX-AIM and appropriate

programs for a faculty which is interested in many aspects of drug design and drug action, ranging from physical chemistry to purely biological studies. He has been encouraged to use GUEST access to explore CONGEN, although he has taken no action up to the present time.

We have cases where requests for GUEST access had to be denied due to system loading considerations. We made these decisions according to the extent to which the requested use would fit within the research guidelines of SUMEX/AIM and our own stated criteria from the 1973 proposal to NIH. In one case, for instance, the use was for an individual's report on potential educational uses of CONGEN.

FUNDING STATUS

The DENDRAL project is in its sixth year of NIH funding through the BRB (Grant RR-00612). For the period 8/1/75 - 7/31/76 the total (direct costs) amount awarded was \$240,967. After nine months of the seventh year the project cease to be supported by the current grant: a competing renewal application will be submitted June 1, 1976. For the nine months period 8/1/76 - 4/30/77 the total (direct costs) amount awarded is \$210,778.

INTERACTIONS WITH THE SUMEX-AIM RESOURCE

The research summary above described several ways in which we see the DENDRAL programs helping biomedical scientists. See Part 3 for a list of persons with whom we have actively collaborated. One of the major goals of the research is to extend the usefulness of the programs for just such persons.

The SUMEX-AIM community is an exciting and productive collection of projects and individuals who contribute in many ways to the progress of all projects in the community. Our programming in INTERLISP, SAIL and FORTRAN, for example, is speeded considerably by the ready availability of expert programmers from many projects. We have shared ideas about intelligent interfaces between programs and users with members of the MYCIN, X-Ray Crystallography and MOLGEN projects. Perhaps the most used and most useful means of communication is the SNDMSG program on SUMEX. It is much more efficient than campus mail and much less intrusive, as well as more efficient for multiple messages, than the telephone. We are cooperating with the SUMEX staff on the Bulletin Board facility, which will be another efficient means of communicating, especially when the sender of a message is not certain who the receivers should be. (It will allow potential receivers to say what they are interested in and notify them of relevant bulletins, without the sender making an explicit distribution list).

The SUMEX-AIM staff is the most professional computer facility staff we have worked with over the ten year life of the DENDRAL project. The very low amount of unscheduled downtime is a direct indication of their professional attitude and abilities. Less measurably, the helpfulness of the staff also translates directly into increased productivity for

DENDRAL. There have been numerous instances of the SUMEX staff answering our questions immediately and fixing errors in system programs for us as quickly as we could expect.

As the system becomes more heavily loaded, we notice longer and longer delays in computer response time. This is the one major criticism voiced by DENDRAL project members. Many of these persons have changed their work habits to conform to the lighter loading between midnight and 5:00 because they cannot get any significant computing done during the day.

SUMMARY OF PUBLICATIONS

- (1) J. Lederberg, "DENDRAL-64 - A System for Computer Construction, Enumeration and Notation of Organic Molecules as Tree Structures and Cyclic Graphs", (technical reports to NASA, also available from the author and summarized in (12)). (1a) Part I. Notational algorithm for tree structures (1964) CR.57029 (1b) Part II. Topology of cyclic graphs (1965) CR.68898 (1c) Part III. Complete chemical graphs; embedding rings in trees (1969)
- (2) J. Lederberg, "Computation of Molecular Formulas for Mass Spectrometry", Holden-Day, Inc. (1964).
- (3) J. Lederberg, "Topological Mapping of Organic Molecules", Proc. Nat. Acad. Sci., 53:1, January 1965, pp. 134-139.
- (4) J. Lederberg, "Systematics of organic molecules, graph topology and Hamilton circuits. A general outline of the DENDRAL system." NASA CR-48899 (1965)
- (5) J. Lederberg, "Hamilton Circuits of Convex Trivalent Polyhedra (up to 18 vertices), Am. Math. Monthly, May 1967.
- (6) G. L. Sutherland, "DENDRAL - A Computer Program for Generating and Filtering Chemical Structures", Stanford Artificial Intelligence Project Memo No. 49, February 1967.
- (7) J. Lederberg and E. A. Feigenbaum, "Mechanization of Inductive Inference in Organic Chemistry", in B. Kleinmuntz (ed) Formal Representations for Human Judgment, (Wiley, 1968) (also Stanford Artificial Intelligence Project Memo No. 54, August 1967).
- (8) J. Lederberg, "Online computation of molecular formulas from mass number." NASA CR-94977 (1968)
- (9) E. A. Feigenbaum and B. G. Buchanan, "Heuristic DENDRAL: A Program for Generating Explanatory Hypotheses in Organic Chemistry", in Proceedings, Hawaii International Conference on System Sciences, B. K. Kinariwala and F. F. Kuo (eds), University of Hawaii Press, 1968.
- (10) B. G. Buchanan, G. L. Sutherland, and E. A. Feigenbaum, "Heuristic DENDRAL: A Program for Generating Explanatory Hypotheses in Organic

- Chemistry". In Machine Intelligence 4 (B. Meltzer and D. Michie, eds) Edinburgh University Press (1969), (also Stanford Artificial Intelligence Project Memo No. 62, July 1968).
- (11) E. A. Feigenbaum, "Artificial Intelligence: Themes in the Second Decade". In Final Supplement to Proceedings of the IFIP68 International Congress, Edinburgh, August 1968 (also Stanford Artificial Intelligence Project Memo No. 67, August 1968).
- (12) J. Lederberg, "Topology of Molecules", in The Mathematical Sciences - A Collection of Essays, (ed.) Committee on Support of Research in the Mathematical Sciences (COSRIMS), National Academy of Sciences - National Research Council, M.I.T. Press, (1969), pp. 37-51.
- (13) G. Sutherland, "Heuristic DENDRAL: A Family of LISP Programs", to appear in D. Bobrow (ed), LISP Applications (also Stanford Artificial Intelligence Project Memo No. 80, March 1969).
- (14) J. Lederberg, G. L. Sutherland, B. G. Buchanan, E. A. Feigenbaum, A. V. Robertson, A. M. Duffield, and C. Djerassi, "Applications of Artificial Intelligence for Chemical Inference I. The Number of Possible Organic Compounds: Acyclic Structures Containing C, H, O and N". Journal of the American Chemical Society, 91:11 (May 21, 1969).
- (15) A. M. Duffield, A. V. Robertson, C. Djerassi, B. G. Buchanan, G. L. Sutherland, E. A. Feigenbaum, and J. Lederberg, "Application of Artificial Intelligence for Chemical Inference II. Interpretation of Low Resolution Mass Spectra of Ketones". Journal of the American Chemical Society, 91:11 (May 21, 1969).
- (16) B. G. Buchanan, G. L. Sutherland, E. A. Feigenbaum, "Toward an Understanding of Information Processes of Scientific Inference in the Context of Organic Chemistry", in Machine Intelligence 5, (B. Meltzer and D. Michie, eds) Edinburgh University Press (1970), (also Stanford Artificial Intelligence Project Memo No. 99, September 1969).
- (17) J. Lederberg, G. L. Sutherland, B. G. Buchanan, and E. A. Feigenbaum, "A Heuristic Program for Solving a Scientific Inference Problem: Summary of Motivation and Implementation", Stanford Artificial Intelligence Project Memo No. 104, November 1969.
- (18) C. W. Churchman and B. G. Buchanan, "On the Design of Inductive Systems: Some Philosophical Problems". British Journal for the Philosophy of Science, 20 (1969), pp. 311-323.
- (19) G. Schroll, A. M. Duffield, C. Djerassi, B. G. Buchanan, G. L. Sutherland, E. A. Feigenbaum, and J. Lederberg, "Application of Artificial Intelligence for Chemical Inference III. Aliphatic Ethers Diagnosed by Their Low Resolution Mass Spectra and NMR Data". Journal of the American Chemical Society, 91:26 (December 17, 1969).
- (20) A. Buchs, A. M. Duffield, G. Schroll, C. Djerassi, A. B. Delfino, B. G. Buchanan, G. L. Sutherland, E. A. Feigenbaum, and J.

- Lederberg, "Applications of Artificial Intelligence For Chemical Inference. IV. Saturated Amines Diagnosed by Their Low Resolution Mass Spectra and Nuclear Magnetic Resonance Spectra", *Journal of the American Chemical Society*, 92, 6831 (1970).
- (21) Y.M. Sheikh, A. Buchs, A.B. Delfino, G. Schroll, A.M. Duffield, C. Djerassi, B.G. Buchanan, G.L. Sutherland, E.A. Feigenbaum and J. Lederberg, "Applications of Artificial Intelligence for Chemical Inference V. An Approach to the Computer Generation of Cyclic Structures. Differentiation Between All the Possible Isomeric Ketones of Composition C₆H₁₀O", *Organic Mass Spectrometry*, 4, 493 (1970).
- (22) A. Buchs, A.B. Delfino, A.M. Duffield, C. Djerassi, B.G. Buchanan, E.A. Feigenbaum and J. Lederberg, "Applications of Artificial Intelligence for Chemical Inference VI. Approach to a General Method of Interpreting Low Resolution Mass Spectra with a Computer", *Chem. Acta Helvetica*, 53, 1394 (1970).
- (23) E.A. Feigenbaum, B.G. Buchanan, and J. Lederberg, "On Generality and Problem Solving: A Case Study Using the DENDRAL Program". In *Machine Intelligence 6* (B. Meltzer and D. Michie, eds.) Edinburgh University Press (1971). (Also Stanford Artificial Intelligence Project Memo No. 131.)
- (24) A. Buchs, A.B. Delfino, C. Djerassi, A.M. Duffield, B.G. Buchanan, E.A. Feigenbaum, J. Lederberg, G. Schroll, and G.L. Sutherland, "The Application of Artificial Intelligence in the Interpretation of Low-Resolution Mass Spectra", *Advances in Mass Spectrometry*, 5 (1971), 314.
- (25) B.G. Buchanan and J. Lederberg, "The Heuristic DENDRAL Program for Explaining Empirical Data". In proceedings of the IFIP Congress 71, Ljubljana, Yugoslavia (1971). (Also Stanford Artificial Intelligence Project Memo No. 141.)
- (26) B.G. Buchanan, E.A. Feigenbaum, and J. Lederberg, "A Heuristic Programming Study of Theory Formation in Science." In proceedings of the Second International Joint Conference on Artificial Intelligence, Imperial College, London (September, 1971). (Also Stanford Artificial Intelligence Project Memo No. 145.)
- (27) Buchanan, B. G., Duffield, A.M., Robertson, A.V., "An Application of Artificial Intelligence to the Interpretation of Mass Spectra", *Mass Spectrometry Techniques and Appliances*, Edited by George W. A. Milne, John Wiley & Sons, Inc., 1971, p. 121-77.
- (28) D.H. Smith, B.G. Buchanan, R.S. Engelmores, A.M. Duffield, A. Yeo, E.A. Feigenbaum, J. Lederberg, and C. Djerassi, "Applications of Artificial Intelligence for Chemical Inference VIII. An approach to the Computer Interpretation of the High Resolution Mass Spectra of Complex Molecules. Structure Elucidation of Estrogenic Steroids", *Journal of the American Chemical Society*, 94, 5962-5971 (1972).

- (29) B.G. Buchanan, E.A. Feigenbaum, and N.S. Sridharan, "Heuristic Theory Formation: Data Interpretation and Rule Formation". In Machine Intelligence 7, Edinburgh University Press (1972).
- (30) Lederberg, J., "Rapid Calculation of Molecular Formulas from Mass Values". Jnl. of Chemical Education, 49, 613 (1972).
- (31) Brown, H., Masinter L., Hjelmeland, L., "Constructive Graph Labeling Using Double Cosets". Discrete Mathematics, 7 (1974), 1-30. (Also Computer Science Memo 318, 1972).
- (32) B. G. Buchanan, Review of Hubert Dreyfus' "What Computers Can't Do: A Critique of Artificial Reason", Computing Reviews (January, 1973). (Also Stanford Artificial Intelligence Project Memo No. 181)
- (33) D. H. Smith, B. G. Buchanan, R. S. Engelmores, H. Adlercreutz and C. Djerassi, "Applications of Artificial Intelligence for Chemical Inference IX. Analysis of Mixtures Without Prior Separation as Illustrated for Estrogens". Journal of the American Chemical Society 95, 6078 (1973).
- (34) D. H. Smith, B. G. Buchanan, W. C. White, E. A. Feigenbaum, C. Djerassi and J. Lederberg, "Applications of Artificial Intelligence for Chemical Inference X. Intsum. A Data Interpretation Program as Applied to the Collected Mass Spectra of Estrogenic Steroids". Tetrahedron, 29, 3117 (1973).
- (35) B. G. Buchanan and N. S. Sridharan, "Rule Formation on Non-Homogeneous Classes of Objects". In proceedings of the Third International Joint Conference on Artificial Intelligence (Stanford, California, August, 1973). (Also Stanford Artificial Intelligence Project Memo No. 215.)
- (36) D. Michie and B.G. Buchanan, "Current Status of the Heuristic DENDRAL Program for Applying Artificial Intelligence to the Interpretation of Mass Spectra". August, 1973. To appear in Computers for Spectroscopy (ed. R.A.G. Carrington) London: Adam Hilger. Also: University of Edinburgh, School of Artificial Intelligence, Experimental Programming Report No. 32 (1973).
- (37) H. Brown and L. Masinter, "An Algorithm for the Construction of the Graphs of Organic Molecules", Discrete Mathematics, 8(1974), 227. (Also Stanford Computer Science Dept. Memo STAN-CS-73-361, May, 1973)
- (38) D.H. Smith, L.M. Masinter and N.S. Sridharan, "Heuristic DENDRAL: Analysis of Molecular Structure," Proceedings of the NATO/CNNA Advanced Study Institute on Computer Representation and Manipulation of Chemical Information (W. T. Wipke, S. Heller, R. Feldmann and E. Hyde, eds.) John Wiley and Sons, Inc., 1974.
- (39) R. Carhart and C. Djerassi, "Applications of Artificial Intelligence for Chemical Inference XI: The Analysis of C13 NMR Data for Structure Elucidation of Acyclic Amines", J. Chem. Soc. (Perkin II), 1753 (1973).

- (40) L. Masinter, N.S. Sridharan, R. Carhart and D.H. Smith, "Application of Artificial Intelligence for Chemical Inference XII: Exhaustive Generation of Cyclic and Acyclic Isomers". Journal of the American Chemical Society, 96 (1974), 7702. (Also Stanford Artificial Intelligence Project Memo No. 216.)
- (41) L. Masinter, N.S. Sridharan, R. Carhart and D.H. Smith, "Applications of Artificial Intelligence for Chemical Inference. XIII. Labeling of Objects having Symmetry". Journal of the American Chemical Society, 96 (1974), 7714.
- (42) N.S. Sridharan, Computer Generation of Vertex Graphs, Stanford CS Memo STAN-CS-73-381, July, 1973.
- (43) N.S. Sridharan, et.al., A Heuristic Program to Discover Syntheses for Complex Organic Molecules, Stanford CS Memo STAN-CS-73-370, June, 1973. (Also Stanford Artificial Intelligence Project Memo No. 205.)
- (44) N.S. Sridharan, Search Strategies for the Task of Organic Chemical Synthesis, Stanford CS Memo STAN-CS-73-391, October, 1973. (Also Stanford Artificial Intelligence Project Memo No. 217.)
- (45) R. G. Dromey, B. G. Buchanan, J. Lederberg and C. Djerassi, "Applications of Artificial Intelligence for Chemical Inference. XIV. A General Method for Predicting Molecular Ions in Mass Spectra". Journal of Organic Chemistry, 40 (1975), 770.
- (46) D. H. Smith, "Applications of Artificial Intelligence for Chemical Inference. XV. Constructive Graph Labelling Applied to Chemical Problems. Chlorinated Hydrocarbons". Analytical Chemistry, in press (to appear May or June, 1975).
- (47) R. E. Carhart, D. H. Smith, H. Brown and N. S. Sridharan, "Applications of Artificial Intelligence for Chemical Inference. XVI. Computer Generation of Vertex Graphs and Ring Systems". Journal of Chemical Information and Computer Science (formerly Journal of Chemical Documentation), in press (to appear in May, 1975).
- (48) R. E. Carhart, D. H. Smith, H. Brown and C. Djerassi, "Applications of Artificial Intelligence for Chemical Inference. XVII. An Approach to Computer-Assisted Elucidation of Molecular Structure". Journal of the American Chemical Society, submitted for publication.
- (49) B. G. Buchanan, "Scientific Theory Formation by Computer." To appear in Proceedings of NATO Advanced Study Institute on Computer Oriented Learning Processes, 1974, Bonas, France.
- (50) E. A. Feigenbaum, "Computer Applications: Introductory Remarks," in Proceedings of Federation of American Societies for Experimental Biology, Vol. 33, No. 12 (Dec., 1974) 2331-2332.
- (51) S. Hammerum and C. Djerassi, "Mass Spectrometry in Structural and Stereochemical Problems - CCXLIV; The Influence of Substituents and

Stereochemistry on the Mass Spectral Fragmentation of Progesterone." Tetrahedron (accepted for publication), 1975.

- (52) S. Hammerum and C. Djerassi, "Mass Spectrometry in Structural and Stereochemical Problems CCXLV. The Electron Impact Induced Fragmentation Reactions of 17-Oxygenated Progesterones." Steroids (submitted for publication).
- (53) H. Brown, "Molecular Structure Elucidation III." Submitted for publication to SIAM Journal on Computing.
- (54) R. Davis and J. King, "Overview of Production Systems" To appear in Machine Representation of Knowledge, Proceedings of the NATO ASI Conference, July, 1975.
- (55) B. G. Buchanan, "Applications of Artificial Intelligence to Scientific Reasoning." In Proceedings of Second USA-Japan Computer Conference, August, 1975.
- (56) R. E. Carhart, S. M. Johnson, D. H. Smith, B. G. Buchanan, R. G. Dromey, J. Lederberg, "Networking and a Collaborative Research Community: A Case Study Using the DENDRAL Program," to appear in Computing Networking in Chemistry, Peter Lykos, ed., American Chemical Society Symposium Series, No. 19, 1975.
- (57) D. H. Smith (Paper XVIII) "The Scope of Structural Isomerism". Journal of Chemical Information and Computer Sciences, 15, 203 (1975).
- (58) B. G. Buchanan, D. H. Smith, W. C. White, R. Gritter, E. A. Feigenbaum, J. Lederberg and C. Djerassi, "Applications of Artificial Intelligence for Chemical Inference. XXII. Automatic Rule Formation in Mass Spectrometry by Means of the Meta-DENDRAL Program." Submitted to Journal of the American Chemical Society.
- (59) E. H. Shortliffe, R. Davis, S. G. Axline, B. G. Buchanan, C. C. Green and S. N. Cohen, "Computer-Based Consultations in Clinical Therapeutics: Explanation and Rule Acquisition Capabilities of the MYCIN System." Computers and Biomedical Research 8, 303-320 (1975).
- (60) R. Davis, B. Buchanan and E. Shortliffe, "Production Rules as a Representation for a Knowledge-Based Consultation Program", accepted for publication by Artificial Intelligence. (Also Stanford Artificial Intelligence Project Memo No. AIM-266.)

IV.A.1.b MYCIN PROJECT

Computer Based Consultation in Clinical Therapeutics

Prof. S. Cohen, M.D. (Pharmacology) and
Dr. B. Buchanan (Computer Science)

(Grant HEW HSO-1544-02, 3 years, \$163,965 this year)

Introduction

This report offers a review of the progress made by the MYCIN project over the past year. To provide some background, we start by describing the system's basic task, and document its significance. This is followed by a description of the way knowledge is represented and used in the system, and a brief discussion of the advantages of the representation we have chosen. The progress report follows this, detailing the accomplishments of the past twelve months, and spells out the plans for the coming year.

Background

The ultimate aim of the MYCIN project has been to develop a computer-based system to which physicians will refer for antimicrobial therapy advice. One primary consideration for the system has been its level of performance. In order to provide a tool which would actually be useful, and be used in the clinical setting, we have to provide a system which displays a high level of competence in its field. Clinicians must have confidence in a program's ability before they will be willing to use it. A second consideration has been the ability of the system to explain its reasoning. Since clinicians are not likely to accept such a system unless they can understand why the recommended therapy has been selected, the system has to do more than give dogmatic advice. It is also important to let it explain its recommendations when queried, and to do so in terms that suggest to the physician that the program approaches problems in much the same way that he does. This permits the user to validate the program's reasoning, and to reject the advice if he feels that a crucial step in the decision process cannot be justified. It also gives the program an inherent instructional capability, allowing the physician to learn from each consultation session. Third, we feel it is desirable that an expert in infectious disease therapy who notes omissions or errors in the program's reasoning should be able to augment or correct the knowledge base so that future consultations will not repeat the same mistakes. The system should therefore have some capability for acquiring knowledge via interaction with experts in the field.

Progress towards these goals has been made in development of the MYCIN system, composed of three interrelated modules. The Consultation System uses MYCIN's knowledge base along with patient data entered by the physician to generate therapeutic advice. The Explanation System has the ability to generate a thorough documentation of the motivation for

questions the system asks or of the rationale for conclusions it reaches. Finally, experts may use the Rule Acquisition System to update MYCIN's knowledge base. Together, these three modules give the system a wide range of capabilities for dealing with the problem of advising on diagnosis and therapy selection for infectious disease.

Significance of the problem

The task of therapy selection for infectious disease was chosen because of the demonstrated need for high quality advice in this area. There have been numerous studies detailing the misuse of antibiotics and its resultant cost. One study (reference [2]) indicates that in a recent year, one of every four people in this country were given penicillin, and nearly 90% of these prescriptions were unnecessary. A major evaluation of the antibiotic prescribing habits of a wide range of specialists was reported within the last two months in reference [3]. It indicates that the overall score was only 68% correct, and suggests possible underlying causes. While there are a number of sociological factors which are also significant (e.g. patient pressure for treatment even when none is indicated), the study suggests that causes for the low score range from the fact that physicians may be unfamiliar with generic names for antibiotics, to a lack of knowledge of basic bacteriology, to the fact that they appear to use antibiotics as a substitute for clinical judgment. Problems such as these indicate the need for more (and more accessible) consultants to physicians selecting antimicrobial drugs.

General Approach

To give a general feeling for the way MYCIN works, we present here a brief description of the way knowledge is represented in the program, and indicate how it is used. We also suggest some of the advantages which result from embodying knowledge in the format we have chosen.

All knowledge used by MYCIN during a consultation session is contained in decision rules that have been coded and stored in the machine. The MYCIN Project members have identified approximately 400 such rules during discussions of representative case histories. Each rule consists of a set of preconditions (called a PREMISE) which, if true, permits a conclusion to be made or an action to be taken, according to the ACTION part of the rule. Figure 1 below shows one such rule.

If

- 1) the stain of the organism is gramnegative, and
- 2) the morphology of the organism is rod, and
- 3) the aerobicity of the organism is aerobic,

Then

there is suggestive evidence (.6) that the identity of the organism is bacteroides.

RULE124

Figure 1

The system uses its collection of rules to make its conclusions. If, for instance, it is attempting to determine the identity of an organism which is causing an infection, it retrieves the entire list of rules which, like the one above, conclude about identity. It then attempts to determine the truth of the premise of the first rule on the list by evaluating in turn each of the clauses of its premise. Thus, for the rule above, the first thing to find out is the gramstain. If this is already available in the data base, the program retrieves it from there. If not, gramstain becomes the new goal, we retrieve all rules which conclude about it, and try to use each of them to obtain the value of the gramstain. If, after trying all the rules on the list, the answer still has not been discovered, the program asks the user. The rules thus "unwind" to produce a succession of goals, and it is the attempt to achieve each goal that drives the consultation. Figure 2 below gives a graphical view of this process. (A more complete description of the program's operation can be found in reference [1]).

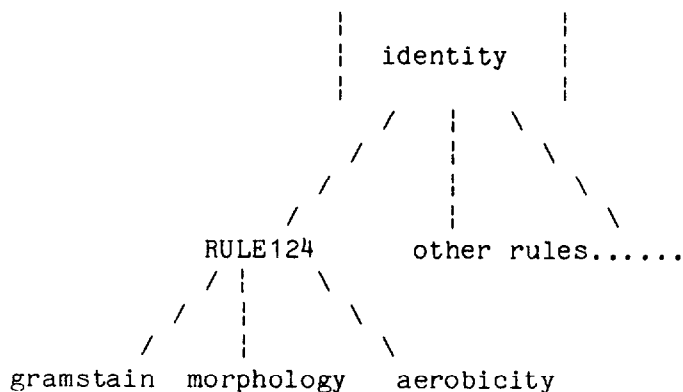


Figure 2

Many of the system's important capabilities are made possible by way knowledge is represented in rules like the one in Figure 1. Such rules offer modular "chunks" of knowledge about the domain, represented in a form that is comprehensible to the clinician. For instance, if the system

is asked "How did you determine the identity of ORGANISM-1? ", it answers by displaying each of the rules which were actually used, in the format shown above. This is something which the clinician can readily understand, and it provides a far more comprehensible explanation than would be possible if the program were to use a statistical approach to diagnosis.

It also means that the expert clinician can offer new "chunks" of knowledge, by expressing them in this same form. He can therefore help to make the program more competent, without having to know anything about computer programming.

There are several other interesting and important benefits gained from the approach we have chosen. These are explained in more detail in reference [1].

Progress report

General objectives and goals during the previous year

During the past year's work on the MYCIN system our goals have been (i) to increase the competence and broaden the scope of the system's therapeutic advice; (ii) to provide additional features to increase utility and performance of the system in the clinical setting; (iii) to develop further the system's collection of user-oriented features to make it easier for novices to use; (iv) to make it possible for an infectious diseases expert (who may know nothing about programming) to interact with and educate the program directly; (v) to develop new techniques to deal with the technical problems of managing a large and growing system; and (vi) to design and execute a formal evaluation study to measure the system's performance. We consider each of these in turn.

Competence and scope

One of the major accomplishments of the past year was the extension of MYCIN to cover diagnosis and therapy for meningitis infections. Over 100 new rules were added to provide this capability. This has proved to be an especially useful new domain to investigate because it has presented several new challenges. In particular, meningitis requires the ability to deal with a disease that is often diagnosed on clinical grounds before any specific microbiologic evidence is available. We have thus found it necessary to consider a larger range of clinical factors. This has resulted in a system which has a broader picture of the whole patient, and thus directly confronts one of the concerns about earlier versions of the system. The system has also become more robust, because it requires less hard microbiologic data, and is thus less sensitive to inaccurate laboratory reports. Like expert clinicians, it is now alert to the possible existence of anomalous data.

The broader range of expertise also means that the MYCIN can begin to play a much more effective role in the clinical setting. Another early concern was that a system with too narrow a range of capabilities demands a great deal of judgment before it is even used. Thus, if MYCIN could

deal only with bacteremia, the user would need to decide that the patient indeed had a bacteremia before he could use the system. By giving MYCIN the ability to diagnose and treat a broader range of infections, we allow it to become useful at a much earlier stage in a patient's clinical course.

Other contributions to the system's competence came from the expansion of the knowledge base to include information about normal flora for a wide range of culture sites. MYCIN can now usually distinguish between normal and pathological flora, and can hence decide more precisely whether to treat.

We have also investigated the addition of some widely applicable routines for computing drug dose in renal failure. These have been developed by independent investigators, but are available to us and could prove to be extremely useful. Our system currently issues warnings simply to modify dosage in renal failure. Since the problem of determining renal status and the proper adjustment of drug dose is difficult, customized drug dosage recommendations will be an important addition to the power of the system.

There have also been significant improvements in the system's ability to handle organism genus and species. The problem requires that the system be able to deal with varying degrees of specificity; at times it can deduce both genus and species, and at others only the genus. Yet it must be able to prescribe correctly in all cases. A fundamental review of the problem has resulted in the addition of a number of new rules which handle the problem comprehensively and uniformly.

Additional clinical features

Several new features have been added to the system in anticipation of its use on the wards. MYCIN now keeps continuous statistics on the use of individual rules from its knowledge base. This will help us to monitor long term performance, to study interrelationships between rules, and perhaps detect inconsistencies or gaps in the knowledge base.

Also looking ahead, we have designed an "on-line" evaluation. At the end of each consultation, the system will ask a few questions about quality of performance, to get some feedback from the clinicians who are actually using it. This interchange will be very brief to avoid being a burden to the user, but will offer a very important form of instant criticism from our users.

User-oriented features

Several "human engineering" capabilities have been improved over the past year. For instance, the system's handling of questions asked by the physician has been made more powerful. This was achieved by improving our handling of English text, and by a comprehensive review of the kinds of questions that are asked. The system can now answer a broader range of questions, and, in particular, can explain why it did not take a specific

action, as well as why other conclusions were reached. Capabilities like these are very important in allowing our local clinical experts to discover the program's rationale for its actions. They can then evaluate its line of reasoning, and suggest any necessary changes.

We are also engaged in a comprehensive effort to put all of the system's deductive actions into rules. Some important steps were previously performed by blocks of code, and hence could not easily be explained by the system. We have begun to reformulate the process in terms of rules. This will permit the system to be more specific about the source of its drug recommendations.

We have also added several new capabilities to provide more convenient use of the system in anticipation of its use on the wards. Among these are the ability for the user to type a comment about system performance at any time during the consultation. His comment is recorded in a special file which is periodically reviewed by our medical staff. This is in addition to the "on line" evaluation described above, and allows the user to offer any comment which he may feel is relevant.

We also have a parallel ability to report problems. The user can indicate that the system has "broken down" in some way, and is invited to describe the problem. His description is saved along with a copy of the program, so that our systems programmers can fix it later.

Linking the expert and the program

We have recently implemented a prototype version of a "bridge" between the clinical expert in infectious disease and the program, which will allow the expert to "teach" the program directly. Formerly, the expert's comments on the system's performance were given to a programmer, who then made the relevant changes to MYCIN. Now the expert can himself begin to discover the source of many problems, and can indicate the necessary rules. The dialogue is carried out in English, and requires no knowledge of programming.

Technical issues

Several changes in the structure of the program have made it easier to deal with the large and constantly changing knowledge base. In general, we are faced with the challenge of keeping the system's size within well specified limits, and have devoted some effort to insure that it remains sufficiently compact. We have, for instance, separated MYCIN's dictionary of English words from the rest of the system. This not only reduces the space requirement considerably, but has an additional benefit of making it easier to update the dictionary as the system grows.

There have always been extensive "self-documenting" capabilities in the system that is, MYCIN can supply instructions and helpful information if the user is confused at any point. We have recently improved the handling of this feature so that it is both faster and requires less space.

Formal evaluation study

A major undertaking this year has been the design and execution of a formal evaluation of the system's performance. The basic idea was to give the same clinical data to both MYCIN and a set of recognized experts in infectious disease therapy, to compare their judgments, and to ask the experts to evaluate MYCIN's performance. We began by designing a form that would allow us to separate the variables requiring analysis. We attempted to determine whether MYCIN (1) asks too many or too few questions, (2) correctly determines which infections require treatment, (3) correctly identifies the organisms that may be causing the relevant infections, and (4) adequately selects therapy to cover for the relevant organisms. The form was designed to be maximally informative, but very simple to complete. It interweaves a sample consultation with questions to the expert, and asks him to record his own opinions regarding the patient and appropriate therapy. It was tested first in a pre-evaluation trial run, with five patients evaluated by three local Fellows in the Division of Infectious Disease at Stanford.

For the formal study, fifteen patients were selected according to strictly defined criteria. For each of these patients we prepared a 1-2 page clinical summary and made copies of relevant material from the patient's chart. This information was used to obtain therapeutic advice for each of them. Questions posed by MYCIN were answered solely on the basis of information collected from patient records at the time of the first positive blood culture, to simulate actual clinical use of the system. These consultations were integrated into the forms and sent along with the clinical data to ten experts in infectious disease therapy.

We had decided some time ago that the introduction of the system onto the wards for experimental use would be predicated on a successful outcome of this evaluation. Thus, while we had originally expected to begin use on the wards some time this year, the large amount of work involved in carrying out the evaluation has delayed us. We feel quite strongly that premature introduction of the program would be unwise, since it would almost surely lead to reduced acceptance by the clinical staff. Upon the return of the evaluation forms in mid- to late-March we shall have sufficient data to determine not only the current level of MYCIN's performance, but also the degree of agreement among the experts themselves. By sending five of the ten evaluation packets to experts in other parts of the country, we are also attempting to determine to what extent MYCIN reflects clinical judgments that may be peculiar to the Stanford environment.

In summary, our work in the past year has focussed on broadening the scope of clinical competence of the system, and on evaluating its performance. In anticipation of the use of MYCIN on the wards, we have added and strengthened many features, to insure that the program is maximally useful to the clinician who seeks advice.

Plans for the coming year

There are a number of major projects planned for the coming year.

There will, for instance, be extensive testing of the new meningitis rules, to insure both that their performance is satisfactory, and that there are no unforeseen side effects on the rest of the system. We plan also to begin work on a knowledge base for pneumonia as the next step in increasing the program's scope. The introduction of the system onto the wards will give us valuable experience on a wide range of cases, and provide a basis for on-going monitoring and evaluation of performance.

We plan also to restructure part of the program's approach to requesting information from the user. Our current technique has developed a small number of technical problems, the most important of which involves the order in which questions are asked. By reorganizing some aspects of the control structure slightly, we will be able to solve all of the technical problems. From the user's point of view the system will continue to function as before, but at the start of a consultation it will ask a number of questions to get a global picture of the current case. This offers the additional, unanticipated, advantage of making interaction with the system seem more natural to the user who is used to presenting the consultant with a brief overview of the case at hand.

We have recently discovered an increasing need for the ability to use information about one infection to conclude things about another, as for instance when one infection has clinical implications about another. We plan in the coming year to implement this capability in a quite general fashion, so that we can deal with interrelationships of infections, cultures, organisms, and so on.

As the program becomes available on the ward, it will become more important to be able to tell the system about new information which may arrive several days after the initial consultation. Thus, as new test results arrive, or as the patient's condition changes, it should be possible to add the new information, and obtain updated recommendations. We plan to implement this too in the coming year.

Finally, since one of our fundamental tasks is the assembly of large amounts of knowledge of infectious disease diagnosis and therapy, we plan to develop further the prototype "bridge" which links the medical expert with MYCIN. Our current version lacks in particular many of the "human-engineering" aspects which are so extensively developed in the rest of the system. We foresee an important acceleration of this process of knowledge gathering when it becomes easy for an expert by himself to make significant changes to the knowledge base.

References

- [1] Davis R, Buchanan B, Shortliffe E H, Production Rules as a Representation for a Knowledge-based Consultation System, A.I. Memo 266, Stanford University, November 1975. (submitted to Artificial Intelligence).
- [2] Kagan B M, Fannin S L, Bardie F, Spotlight on Antimicrobial Agents - 1973, JAMA, 226, 3 (October 1973) pp 306-310.

- [3] Neu H C, Howrey S P, Testing the Physician's Knowledge of Antibiotic use, NEJM, 293, 25, (18 Dec 75), pp 1291-5.

The MYCIN Project - List of Publications

- [1] Scott A C, Clancey W, Davis R, Shortliffe E H, Explanation capabilities of knowledge based production systems (in preparation).
- [2] Shortliffe E H, Davis R, Some considerations for the implementation of knowledge-based expert systems, SIGART Newsletter, 55:9-12, December 1975.
- [3] Shortliffe E H, Computer-Based Medical Consultations: MYCIN, (adaptation of thesis), American Elsevier, New York, 1976
- [4] Davis R, Buchanan B, Shortliffe E H, Production rules as a representation for a knowledge-based consultation system A.I. Memo 266, Stanford University, November 1975. (submitted to Artificial Intelligence).
- [5] Davis R, King J J, An Overview of Production Systems Machine Representations of Knowledge, Proceedings of NATO ASI Conference, to appear, Spring 1976. (Also A.I. Memo 271, Stanford University, October 1975).
- [6] Shortliffe E H, Judgmental knowledge as a basis for computer-assisted clinical decision making, Proceedings of the 1975 International Conference on Cybernetics and Society, pp 256-7, September 1975.
- [7] Shortliffe E H, Axline S, Buchanan B G, Davis R, Cohen S, A computer-based approach to the promotion of rational clinical use of antimicrobials, International Symposium on Clinical Pharmacy and Clinical Pharmacology, Sept 1975, Boston, Mass. (invited paper)
- [8] E H Shortliffe, R Davis, S G Axline, B G Buchanan, C C Green, S N Cohen, Computer-based consultations in clinical therapeutics: explanation and rule acquisition capabilities of the MYCIN system Computers and Biomedical Research, 8:303-320 (August 1975).
- [9] E H Shortliffe and B G Buchanan, A Model of Inexact Reasoning in Medicine, Mathematical Biosciences 23:351-379, 1975.
- [10] Shortliffe E H, Rhame F S, Axline S G, Cohen S N, Buchanan B G, Davis R, Scott A C, Chavez-Pardo R, and van Melle W J, MYCIN: A computer program providing antimicrobial therapy recommendations (abstract only). Presented at the 28th Annual Meeting, Western Society For Clinical Research, Carmel, CA, 6 Feb 1975. Clin. Res. 23:107a (1975). Reproduced in Clinical Medicine, p. 34, August 1975.
- [11] Shortliffe E H, MYCIN: A rule-based computer program for advising physicians regarding antimicrobial therapy selection (abstract only);

Proceedings of the ACM National Congress (SIGBIO Session), p. 739, November 1974. Reproduced in Computing Reviews 16:331 (1975).

- [12] E H Shortliffe, S G Axline, B G Buchanan, S N Cohen, Design considerations for a program to provide consultations in clinical therapeutics, Presented at San Diego Biomedical Symposium 1974 (February 6-8, 1974).
- [13] E H Shortliffe, S G Axline, B G Buchanan, T C Merigan, S N Cohen, An artificial intelligence program to advise physicians regarding antimicrobial therapy, in Computers and Biomedical Research, 6:544-560 (1973).
- [14] Shortliffe, E H, MYCIN: A rule-based computer program for advising physicians regarding Antimicrobial therapy selection, Thesis: Ph.D. in Medical Information Sciences, Stanford University, Stanford CA 409 pages, October 1974. (available from NTIS as document ADA001373)

Funding Status

MYCIN is funded by the Bureau of Health Services Research and Evaluation, and is currently completing the second year of a three year grant (S. Cohen & S. Axline, Principle Investigators). The budget for the current year (6/1/75 - 5/31/76) is \$163,965; a total of \$149,982 is requested for 6/1/76 - 5/31/77. Renewal for the coming year is currently under-going an in-house review. A new 3-year grant request for comparable levels of funding will be submitted in the fall of 1976.

Interaction with SUMEX-AIM Resource

During the past year, we have been contacted by a number of physicians who had read about MYCIN and were interested in trying out the system. Using TYMNET, these physicians in Boston, San Diego, Seattle, Washington D.C., and Atlanta were able to use the SUMEX GUEST account to run consultations on test cases.

MYCIN users are urged to send us comments about the system's performance. [A new "COMMENT" feature in the system allows comments to be entered at any time, without interrupting the consultation, nor even requiring the user to know how to use SNDMSG.] Recent comments from doctors associated with Rutgers Computers in Biomedicine served as very helpful guidelines for making the program's instructions and questions easier for a naive user to understand. Such comments also focus our attention on deficiencies in the program's medical knowledge as well as pointing out programming problems which may exist.

We have continued interaction via SNDMSG and terminal links with members of the DIALOG group, who recently wrote MYCIN-like rules for diagnosing and treating venereal diseases. We have implemented these rules and can modify or add to them as the doctors in Pittsburgh run more consultations to test the validity and completeness of this set of rules.

At a recent 3 day mini-conference, and at weekly meetings, members of the different SUMEX-AIM projects which make up the Heuristic Programming Project at Stanford discuss common problems faced by all the groups, and how each group handles these problems. These discussions have proved very helpful, especially to those projects which are currently in the design stage. Several of the projects have been able to benefit from the work that has been done in MYCIN on designing production rules and explanation capabilities.

Critique of Resource Services

Development of MYCIN has been greatly facilitated by the availability of the Interlisp language and its extensive interactive debugging capability and user-oriented features; in fact, it is doubtful that MYCIN could have developed to its current state without this large-scale interactive resource. However, in recent months its use has been severely limited by the poor response time during peak hours, which effectively prevents the use of MYCIN or Interlisp at such times. In this regard, we have found useful the SUMEX batch facility, which permits us to run some of our non-interactive tasks at times of low system usage.

We are fairly pleased with the availability of disk storage, although its availability may, in the foreseeable future, present some problems. Continuing development of the project makes substantial demands on disk space, since both experimental and publically accessible version of the system must be kept available, as well as a library of patient cases, system dictionaries, and documentation. The archival and retrieval mechanism currently available has proved to be very helpful, and we have made considerable use of it. This, along with careful management of the available space, has made it possible to avoid any problems at this time. As project development continues, however, we anticipate that disk space may become a scarce resource.

One of the outstanding aspects of the facilities at SUMEX continues to be the attitude and competence of its staff and systems people. They seem constantly willing to help with problems and consider suggestions, encouraging a sense of cooperation in the user community. Their on-going development of text editors and other features of the system contribute directly to its utility as a scientific resource.

IV.A.1.c PROTEIN STRUCTURE MODELING PROJECT

Protein Crystallography Project

Prof. J. Kraut and Dr. S. Freer (Chemistry, U. C. San Diego) and
Prof. E. Feigenbaum and Dr. R. Engelmores (Comp. Sci., Stanford)

(Grant NSF DCR74-23461, 2 years, \$88,436 total)

I. Summary of Research Objectives

A. Collaboration, Biomedical Relevance and Technical Goals

The Protein Crystallography Project is a collaboration of two research groups, one at Stanford University, the other at the University of California at San Diego. The Stanford group consists of Edward Feigenbaum, Robert Engelmores, Penny Nii, and, during the current academic year, Carroll Johnson of Oak Ridge National Laboratory. The primary activities are to 1) identify critical tasks in protein structure elucidation which may benefit by the application of AI problem-solving techniques, and 2) design and implement programs to perform those tasks. The UCSD group consists of protein crystallographers: Joseph Kraut, Richard Alden and Stephan Freer. As protein crystallographers, their objective is to seek new techniques that will facilitate the elucidation of the tertiary structure of proteins.

The biomedical relevance of protein structure determination is well known. Solved protein structures have contributed substantially to our understanding of molecular biology, enzymology and immunology.

We have identified two principal areas where we believe the collaboration is of practical and theoretical interest to both protein crystallographers and computer scientists working in AI. The first is the problem of interpreting a 3-dimensional electron density map. The second is the problem of determining a plausible structure in the absence of phase information normally inferred from experimental isomorphous replacement data.

B. Specific Objectives

1. Interpretation of electron density maps

A major challenge in protein crystallography is the interpretation of the crystallographic electron density map. Our goal is to build a knowledge-based system which proposes plausible locations for substructural units consistent with the electron density map, the amino-acid sequence (if known), and physical, chemical and stereochemical constraints. The system attempts to integrate knowledge from three different areas: chemical topology, microstructure and macrostructure. The chemical topology knowledge base is task specific and contains the

known connectivities within the protein structure (i.e., the amino acid sequence, cofactors, disulfide bridges, and coordination bonds to prosthetic groups). The microstructure knowledge base contains known facts about protein molecules (e.g., the molecular geometry of the peptide bond and the amino acid side chains, hydrogen bonding properties, helix forming propensity, etc.). The macrostructure knowledge base contains stereotype templates for the plausible major components of the molecule (e.g., alpha helices and pleated sheets). Our strategy is to isolate an individual molecule within the map, then determine the path of the main chain by a skeletonizing procedure (as is done, for example, by J. Greer, *J. Mol. Biol.*(1974), vol.82, pp. 279-301). We will then parameterize the density along the backbone, identify the most obvious regions (heavy atoms, alpha helices, planar groups) and determine, by region growing and template matching, the identity and position of side chains.

2. Structure Determination in the Absence of Experimental Phase Information

X-ray crystallography is the primary experimental technique for investigating the 3-D structure of molecules. The data so obtained are a collection of intensity measurements at discrete directions with respect to the x-ray beam and crystal axes. These intensities are related to the positions of the atoms in the crystal lattice by a Fourier transformation. Thus, given the structure of the molecule, its orientation with respect to the crystal axes and the symmetry properties which determine the molecular stacking, one can calculate the intensities. However, given the intensities and the unit cell properties, one cannot go the other way. That is, the inverse Fourier transform cannot be calculated because the experiment measures only the amplitudes of the diffracted waves and not their phases -- the classical "phase problem" of x-ray crystallography.

During our first year we have been investigating and implementing various computational techniques for inferring trial structures, in the absence of phase information. Our aim has been to develop a system of computer programs which apply as much general and task-specific knowledge as possible to constrain the search for a plausible structure (or partial structure) consistent with the experimentally measured structure amplitudes.

A procedure well known to x-ray crystallographers investigating the structures of small molecules is that of Patterson search. Recently this technique has been shown to be effective in protein crystallography as well, when there exists a family resemblance between the molecule under investigation and a known protein. Patterson search is basically an image-seeking technique, where one searches for the "Patterson image" of an hypothesized molecular structure in the Patterson map derived from the experimental data. The Patterson function does not require any knowledge of phases.

Patterson search is our primary technique for inferring structures in the absence of experimental phase information. In order to resolve the ambiguities which often arise in Patterson search predictions, we are investigating the use of other knowledge sources, among which are

anomalous dispersion Patterson interpretation, Patterson search in reciprocal space, superposition and Fourier refinement methods. The integration of these diverse knowledge sources is a primary objective of the research.

C. Summary of Project Accomplishments

Our activities during the first year fall into three general categories: augmenting our stockpile of crystallographic computing tools, applying Patterson search methods to a solved protein structure (Cytochrome C2) and an unsolved protein structure (Cytochrome F), and initiating a new research objective.

1. Application of Patterson Search to Cytochrome F

A major goal of our first year of research has been to apply the method of Patterson search, in conjunction with other analytic techniques, to solve a real protein structure. Cytochrome F is an excellent candidate, because (1) phase information is not yet available, and (2) the protein's structure is expected to show a resemblance to other members of the cytochrome family. The current hope is that the family resemblance will be sufficiently strong that the complete structure can be solved by standard refinement techniques after one finds the correct orientation and position of a characteristic substructure. As of this writing a complete solution has not yet been obtained. A considerable effort, described in the remainder of this section has been invested in the pursuit of the correct orientations of the protein in the unit cell. A large number of Patterson search calculations were performed, exploring the effects of variations in the search structures, the selection of search vectors, the choice of measures of fit, and even in the primary data. It now remains to be seen if some of the candidate orientations proposed by the search calculations can be verified by other sources of crystallographic knowledge.

2. Selection of a new research objective.

Shortly after the inception of the project, the two collaborating groups agreed to the need for an additional scientist with an extensive knowledge of crystallography and crystallographic computing, and a serious interest in the application of AI techniques to his field of expertise. We were fortunate to induce Dr. Carroll Johnson of Oak Ridge National Laboratories, who well fulfills these qualifications, to join our project for a one-year period beginning September 1, 1975. His contributions have been instrumental in defining a new task area for the application of AI methodology to protein crystallography. After studying recent work in visual scene analysis, he noted the similarities of that AI application with the crystallographic problem of interpreting a 3-dimensional electron density map, i.e. deriving the coordinates for a trial structure, given the electron density function, the amino-acid sequence and the stereochemical principles and constraints known to apply.

The task so defined contains most if not all of the ingredients for the development of a knowledge-based system, in the mainstream of current AI research. The crystallographer integrates several sources of knowledge -- chemical, stereochemical, crystallographic -- as he builds a model of the protein which is consistent with the given data. He combines this knowledge with a rich set of heuristics for focussing his attention on promising regions of the map, for distinguishing characteristic features, for deciding at what level of detail to stop the interpretation in different regions, and for evaluating competing hypotheses.

The model builder's decision-making process is dynamic and flexible, driven at times by the need to reach specific subgoals, and at other times by the current state of the model or special features of the data. A computer program for interpreting the map will require a control structure which combines both goal-driven and event-driven elements. The design of a suitable control structure, and the implementation of a prototype program for performing the basic interpretive tasks are primary objectives for our second year of research.

3. Assembly of crystallographic computing tools.

With the assistance of several crystallographers around the country we have augmented our collection of crystallographic computing programs and systems (i.e., integrated collections of programs). Those programs we received and/or implemented on SUMEX include:

- a) X-RAY 72. This is a large system of Fortran programs developed by J.M. Stewart (Univ. of Md.) and others. A version written for the DEC-10 at the University of Pittsburgh was kindly furnished to us by Steve Ernst. We have implemented some parts of the system as separate programs, including the Fourier transform, peak finding, and bond length and angle calculating programs.
- b) Sequence-Structure Correlator. A program which predicts alpha helix and pleated sheet regions of a protein molecule from its amino-acid sequence was furnished to us by Ray Salemme (Univ. of Ariz.). The program was subsequently rewritten in both the SAIL and LISP languages. The algorithm is based on the rules developed by Chou and Fasman (Biochemistry, vol. 13, pp. 211-245 (1974)).
- c) Oak Ridge Fast Fourier Program (ORFFP). A system of Fortran programs for generating, analyzing and plotting Fourier maps was obtained from Henry Levy of Oak Ridge and implemented on SUMEX. The plotting segment is the ORTEP program written by Carroll Johnson.
- d) Greer Skeletonization Program. A Fortran program for reducing an electron density map of a protein to a set of connected line segments, following the algorithm proposed by Greer, has been written and is currently being debugged. This program will play a pre-processing role in the map interpretation problem, producing a highly abstracted representation of the map.
- e) Huber Rotation and Translation Search Program. This is a Patterson

search program which, like our PSRCH, computes correlations of two vector sets in the vector space representation. The program, in Fortran, is on file but has not yet been tested at SUMEX.

- f) ROTRAN. These programs were written by B.M. Craven (Univ. of Pittsburgh) and are designed to perform rotational and translational Patterson searching, employing a method developed by Crowther. At present we have only a listing of the Fortran program and instructions for its use.

D. Publications

Feigenbaum, E. A., Engelmores, R. S. and Johnson, C. K., A Correlation Between Crystallographic Computing and Artificial Intelligence Research, submitted for publication in Acta Crystallographica.

II. Interaction with the SUMEX-AIM resource

All program development, and most communications between the two collaborating groups are effected on the SUMEX computer. The UCSD group has a direct connection to SUMEX via the TYMNET computing network (UCSD lost its ARPANET connection during the past year). Routine daily communications now take place using the system's message facility. Program files are equally accessible from Stanford and UCSD, so that either group can construct, edit or exercise the programs. Large data files are transmitted on magnetic tape.

The greatest benefit of the interaction with the SUMEX-AIM resource is the opportunity to share ideas, programming experience and utility programs with other users in the community. The availability of a pool of INTERLISP programmers, for example, has been of great assistance in our initial efforts with the electron density map interpretation task. Members of the SUMEX staff have also been helpful and patient in solving some of the more mundane problems associated with any computational effort (e.g., reading magnetic tapes produced at other computer centers).

IV.A.2 NATIONAL USERS

IV.A.2.a CHEMICAL SYNTHESIS PROJECT

Simulation and Evaluation of Chemical Syntheses (SECS)

W. Todd Wipke
Department of Chemistry
University of California
Santa Cruz, CA. 95064

I. Summary of Research Program

A. Technical Goals

The long range goal of this project is to develop the logical principles of molecular construction and to use these in developing practical computer programs to assist investigators in designing stereospecific syntheses of complex bio-organic molecules. Previously the focus of our work had been to represent as accurately as possible the fundamental chemical transformations and how steric, proximity, and electronic factors affect these reactions, going into great detail even involving analysis of three-dimensional models. The goals for this year focused on developing constraints to help guide the synthesis program in growing the tree of synthetic precursors. We wanted to utilize high level information about symmetry, and stereochemistry to set up strategies defining preferred orderings of making and breaking bonds. We also hoped to completely separate these strategies from the chemical transforms to allow experimenting with changing transforms keeping strategies constant and vice-versa. This separation was also deemed important for ease of maintenance of large transform libraries. Finally we hoped to use these strategies for guiding multistep lookahead so the user could see a sequence developed in the tree at one time.

B. Medical Relevance and Collaboration

The development of new drugs and the study of how drug structure is related to biological activity depends upon the chemist's ability to synthesize new molecules as well as his ability to modify existing structures, e.g., incorporating isotopic labels into biomolecular substrates. The Simulation and Evaluation of Chemical Synthesis (SECS) project aims at assisting the chemist in designing stereospecific syntheses of biologically important molecules. The advantages of this computer approach over a manual approach are manyfold: 1) greater speed in designing a synthesis; 2) freedom from bias of past experience and past solutions; 3) thorough consideration of all possible syntheses using a more extensive library of chemical reactions than any individual can remember; 4) greater capability of the computer to deal with the many

structures which result; and 6) capability of computer to see molecules in graph theoretical sense, free from bias of 2-D projection. SECS was designed to be able to apply any kind of chemical transformation, and because of this generality we see it finding application in biogenesis and metabolism (see section II A below).

C. Progress and Accomplishments

The environment of this project has changed dramatically in the past year with the move from Princeton to Santa Cruz. SECS was moved from a hands-on environment consisting of a KA PDP-10 with an LDS-1 graphics system and standard DEC software to a remote environment with access to a KI PDP-10 through a GT40 graphics terminal where the host monitor system is TENEX. The compatibility package of TENEX considerably eased the problems of conversion. Most problems resulted from differences in file handling, and differences in the Fortran operating system. Subtle problems arose from the fact that our files were organized by tapes and could not simply all be transferred to disk, because of space problems and naming conflicts. SECS was successfully converted to TENEX and the graphical interaction was modified for greater efficiency in our new remote low bandwidth mode of communication.

Progress in developing strategy includes creating a general goal list structure which allows complex logical combinations of goals to be expressed, for example, "(break bond 2 or break bond 3) and use group 1". Thus, instead of one set of "strategic bonds" to be broken, we now can express strategies involving pairs of bonds, or groups or atoms. We have succeeded in isolating strategy from the chemical transforms--strategies can only contain expressions which refer to structural units of the molecule or changes in those units, and may not refer to any transform by name. The transforms have been given "character" which describes the type of structural changes likely to occur if the transform is applied, e.g., cleaves ring, removes group, modifies stereochemistry, etc. The SECS strategy module first sets up standard goal lists based on graph-theoretic heuristics and then allows the user to view and modify the goal lists. In this way the user can place constraints on the syntheses generated, e.g., "don't modify this ring, instead, focus your attention on this part of the molecule." GOALTST was modified to interpret the new complex goals and also to test the achievement or violation of a goal as early as possible. Hence, there is testing before examination of the transform as well as after interpretation of the transform. The net result of this work on strategy is that the user can very closely constrain SECS now to work only in areas which the user decides is worthwhile, consequently fewer precursors are generated which the user would delete.

Significant progress was made in the recognition of symmetry and use of that information in SECS. A general algorithm based on SEMA, our canonical naming algorithm, was developed and implemented for generating the entire symmetry group of the molecule, using the stereochemical graph isomorphism group. We have applied this symmetry knowledge to make SEMA itself more efficient, and have combined it with symmetry of the transforms to make application of a transform generate a non-redundant set of precursors. Thus, if a double bond is introduced into cyclohexane,

only one cyclohexene is generated, not six. This algorithm takes into account all stereocenters in the molecule of both double bonds and saturated carbon. Addition of this algorithm reduces execution time of SECS on certain types of problems by a factor of up to six or more. We have not yet developed heuristics for creating strategies from this symmetry group.

Considerable improvement of aromatic chemistry has resulted from the addition of "character words" to the aromatic transforms and reorganization of the aromatic module. Electronic perception is only performed now if SECS is fairly certain that aromatic chemistry will be used and the user can prevent the MO calculations if he wishes. Work is still progressing on implementation of strategies to control when to apply aromatic transforms based on heuristics derived from an extensive literature study. Many other modifications have been made to improve the human engineering of SECS. Documentation of modules whose authors have left the project is still continuing. Now with an expanding users group, a good user's manual is required and is under revision.

D. Current list of Project Publications

- [1] W.T. Wipke and T. M Dyott, "Use of Ring Assemblies in a Ring Perception Algorithm," J. Chem. Info. and Computer Sci., 15, 140 (1975).
- [2] T.M. Gund, P.v.R. Schleyer, P.H. Gund and W.T. Wipke, "Computer Assisted Graph Theoretical Analysis of Complex Mechanistic Problems in Polycyclic Hydrocarbons. The Mechanism of Diamantane Formation from Various Pentacyclotetradecanes," J. Amer. Chem. Soc., 97, 743 (1975).

Papers in Preparation:

- [1] W.T. Wipke and P. Gund, "Simulation and Evaluation of Chemical Synthesis. Congestion: A Conformation Dependent Function of Steric Environment at a Reaction Center. Application with Torsional Terms to Stereoselectivity of Nucleophilic Additions to Ketones," J. Amer. Chem. Soc.
- [2] W.T. Wipke, G. Birkhead, and T. Brownscombe, "Correlation of Congestion with Stereoselectivity of Olefin Epoxidation."
- [3] W.T. Wipke, C. Still, G. Grethe, T.M. Dyott, and P.E. Friedland, "ALCHEM: A Language for Representing Chemical Transforms. Application to Heterocyclic Chemistry."
- [4] W. Todd Wipke and Hartmut W. Braun, "Graph-theoretical Perception of Molecular Symmetry."
- [5] W.T. Wipke, G. Smith and H. Braun, "SECS-Simulation and Evaluation of Chemical Syntheses: Strategy and Planning," ACS Symposium Proceedings.

E. Funding Status.

IBM Fellowship supporting S. Krishnan (postdoctoral)
\$4000. expires September, 1976

Merck, Sharp and Dohme fellowship supporting Graham Smith
(postdoctoral)
\$1000. expires July, 1976

Sandoz Unrestricted grant to support computer synthesis
\$2000

Proposal submitted 1 Mar 1976 to Division of Research Resources
"Resource-Related Research: Biomolecular Synthesis"
\$391,532 for three years

II. Interactions with the SUMEX-AIM Resource

A. Collaborations and Medical Use of Programs Through Networks. Since SECS only recently has been operating on the SUMEX-AIM resource, collaborations are just beginning. However demonstrations of SECS have been given at the National Cancer Institute and collaboration with Division of Chemical Carcinogenesis to try to use SECS for metabolism of compounds to evaluate carcinogenic activity of metabolites is currently under discussion. The National Library of Medicine toxicology program is also interested in SECS and are planning to access SECS via SUMEX-AIM. Dr. Steve Heller of the EPA and Dr. G.W.A. Milne of the National Heart and Lung Institute are currently exploring the possibility of putting SECS up on the Cyphernetics network. For the past year SECS has been available over the TELENET from First Data Corporation. Squibb, Merck, FMC, American Cyanamid, Pfizer and Searle pharmaceutical companies have used an experimental version of SECS and have provided useful feedback to us about problems they discovered. We expect increasing numbers of academic users will be accessing SECS via SUMEX-AIM as they learn of its availability.

The availability of SECS on SUMEX-AIM has also served health-related research at the University of California, Santa Cruz. For example, model building using the SECS model builders is being performed for Professor Edward Dratz (UCSC) to generate conformations of fatty acids isolated from visual membranes ("Structure and Function of Visual Photoreceptors", EY00175-05), and for Professor Howard Wang (UCSC) to study how conformations of steroids may affect the local anesthetic - membrane interaction ("Role of Membrane Proteins in Local Anesthetic Action," GM22242-01).

B. Cross Fertilization with other SUMEX-AIM projects. The SECS project held joint research group meetings at Stanford with the DENDRAL and AI groups to discuss common problems and research goals. This has been very rewarding since the DENDRAL group has useful experience with symmetry manipulation which SECS was getting into, and the SECS project has useful experience with representing reactions, which DENDRAL/CONGEN was getting into. These joint meetings also let the members meet in person after having met on-line on the network. Last year's AIM

Conference at Rutgers was also a valuable experience, which allowed us to meet people interested in similar problems in different disciplines, and it also caused us to think about what we were doing in research with some new perspectives. We are looking forward to this year's AIM meeting.

We find the SUMEX-AIM network very well human engineered. The ability to leave messages on the network, and to LINK to other users on-line for advice has been extremely useful to us, since we were new to the TENEX operating system. But more than that, we have been able to utilize expertise of others which our group lacked, e.g., Trisha Davis (an undergraduate) has been writing a model builder and display program in SAIL although there is no SAIL expertise in the SECS group--that would not have been possible without the network communication features.

C. Critique of Resource Services. The SECS project finds the SUMEX-AIM staff and community extremely helpful, and anxious to extend themselves to meet our needs. SUMEX provided a leased line and modems to us and provided TYMNET access as well. Were it not for SUMEX, this research effort would have perished since there is no adequate computer facility on-campus.

We do find we are short of disk space and in our grant proposal we have requested funds for a disk drive to place at SUMEX to help resolve this problem. The response time during the day and sometimes even later is poor for interactive graphics, but hopefully the second processor being installed will help alleviate that difficulty. We have an additional problem that it is difficult to transfer files from TENEX to any other PDP-10 with the files retaining their filenames. This problem may also be resolved if we are able to write tapes locally from over the network. Basically we have found that SUMEX-AIM provides a productive and scientifically stimulating environment and we are thankful that we are able to access the resource and participate in its activities.

IV.A.2.b INTERNIST (DIALOG) PROJECT

INTERNIST - Diagnostic Logic Program
Dr. H. Pople and J. Myers, M.D.
University of Pittsburgh

(Grant HEW MB-00144-01, 3 years, \$167,168 last year)

I. SUMMARY OF RESEARCH PROGRAM

A. BACKGROUND AND OBJECTIVES

The principal objective of the MIS laboratory at the University of Pittsburgh is to develop, test, and implement a computer-based diagnostic consultation system for internal medicine. Considerable progress towards this goal had already been made prior to our receipt, in June, 1974, of a three-year \$524,000 grant from the Bureau of Health Resources Development to establish a "Computer Laboratory Health Care Resource" at Pitt. At that time, the medical data base accessed by the internist (formerly DIALOG) program was estimated to comprise approximately twenty-five percent of the major diseases of internal medicine, and a number of case studies had been run illustrating the power of the INTERNIST heuristic process in dealing with a variety of complex clinical problems. Our research plan envisioned a five-year development effort, intended to yield:

- (a) A four-fold expansion of the data base.
- (b) Systematic field testing and evaluation of the system in actual clinical settings.
- (c) Eventual implementation, making INTERNIST available for clinical use on a routine basis.

B. PROGRESS AND ACCOMPLISHMENTS

Shortly after award of the BHRD grant, arrangements were concluded permitting use of the SUMEX-AIM computer resource for INTERNIST research and development activities. Although the SUMEX-AIM computer is of the same genre as the one used in the original INTERNIST development work, differences in the LISP language supported necessitated major conversion efforts.

As mentioned in our last progress report,, the need for rapid access to large data files motivated the design of an interface between the INTERLISP host processor and a set of structured files containing the entire vocabulary and network of associations comprising the INTERNIST data base.

As of June, 1975, conversion of the data base had been completed and

the necessary interfaces had been established to enable INTERNIST diagnostic programs to work with these revised structures. design and implementation of an interactive data entry and editing system, was completed in December, 1975, enabling expansion of the on-line data base to its present size which is approximately 60% complete.

This newly expanded data base is currently being subjected to extensive testing in both typical examples of disease and difficult diagnostic problems. This procedure of systematically checking the entire clinical data base should be completed by late June, 1976. The planned field test and evaluation effort will then commence in early fall.

C. PUBLICATIONS

- [1] Pople, H.E., "Artificial Intelligence Approaches to Computer-based Medical Consultation; Proceedings of IEEE Intercon, 1975, New York.
- [2] Pople, H.E., Myers, J.D., and Miller, R.A., "DIALOG: A Model of Diagnostic Logic for Internal Medicine," Proceedings of Fourth International Joint Conference on Artificial Intelligence, Tbilisi, Georgia, USSR, 1975.

II. INTERACTIONS WITH SUMEX-AIM RESOURCE

Because this year has been largely devoted to system development and checkout activities, there has been no real opportunity to engage in any meaningful collaboration via the communication networks associated with SUMEX-AIM. We fully expect to exploit this attractive feature of the resource, however, during the evaluation and field test studies planned for the coming year.

Concerning the service provided by SUMEX-AIM, our only complaint is the heavy loading during prime hours, which effectively prevents serious use of the INTERNIST diagnostic programs during certain portions of the day. We applaud and eagerly await the advent of the SUMEX-AIM dual processor.

IV.A.2.c HIGHER MENTAL FUNCTIONS MODELING

HIGHER MENTAL FUNCTIONS MODELING (HMF)
Project Summary - 1976Kenneth M. Colby, M.D.
Professor of Psychiatry, UCLA

(NIH MH-27132-01, 2 years, \$67,000 this year)

Introduction.

One of the oldest and newest applications of computers in artificial intelligence is the simulation of human cognitive processes. The Higher Mental Functions project has been modelling belief systems and related psycho-pathological delusional systems for a number of years. The specific goal for the past two years has been to construct, test, and validate a computer simulation of paranoid processes. The development of such a model has clinical implications for the understanding, treatment, and prevention of paranoid disorders.

Recently we have been focussing on the origin of beliefs in belief systems and the criteria by which beliefs are significant to the entity we are modelling; i.e. the motivation for the beliefs. The motivation for an entity's purposive behavior is based in its affect or emotion system. We are currently formulating a theory of the motivational influence of affect on conative (volitional) and cognitive (inferential) processes, with the intent of implementing this background theory in a simulation model. By specifying the underlying theory of motivation we hope to make the theory of paranoia more explicit and the paranoid simulation model more adequate.

The strategy of computer simulation of mental processes can be characterized roughly by three phases: (1) identification and critical description of non-random patterns occurring in the phenomena under study, (2) explanation by postulation of underlying mechanisms which generate, produce, or are responsible for the non-random patterns, and (3) validation by repeated attempts to test the reality of the proposed theory or model. The construction and use of simulation models of mental processes closely parallels model-building in other sciences. An attempt is made to reproduce the relevant features of the patterns under study. This attempt produces simplification and idealization of the phenomena. Simplification implies that only centrally relevant variables are chosen for representation in the model. Idealization implies that exact classes and perfect properties are assumed in the implementation of the model. Still, the model can provide an explanation of underlying mechanisms which is useful in understanding and interpreting the observed patterns of phenomena. Finally, a model can be used in practical situations for prediction, and for providing suggestions to clinicians for potential control and change in the phenomena. Such technological purposes are important for models of mental disorders since the long-range goal of

mental health research is to prevent or reduce conditions of human mental suffering.

Simulations of cognitive processes are difficult for a number of reasons:

- (1) The underlying generative mechanisms of human behavior are inaccessible to direct observation, and must instead be postulated as hypothetical constructs that may (possibly) account for the phenomena.
- (2) A simulation must take into account the rich background of information that a human has available to apply to a contemporary situation.
- (3) Human beings have internal needs which are a function of the immediate past and present, as well as the long-term past of the individual. These needs and past experiences color the human's response to the contemporary situation.
- (4) Human linguistic behavior is the richest source of data for exploring cognitive processes as well as the most complex and therefore desirable behavior to simulate. At the same time, it is difficult due to the variety of behavior possible and the variety of explanations possible for one specific linguistic action.
- (5) Once a simulation model is performing, it is difficult to show the subtleties of the model's generative mechanisms. Instead, some attempt is usually made to reveal the internal workings of the model and appeal to the observers' intuition and/or introspection.

Our overall purpose, then, is to develop theories of human mental processes, specifically psycho-pathological processes, and to implement these theories in computer simulation models. The simulation models help formalize and explicate their associated theories by forcing them into a single notation and requiring the theory-builders to specify the details of the theory. In addition, the models provide a testing ground for validating the theory. On the basis of such theories, we hope to explain the origin of psycho-pathologies and offer principles on which to base treatment and prevention.

Technical goals.

A. Expand the theory of paranoia. The theory implemented in the current model (PARRY2), the humiliation theory, postulates that informational inputs from other people activate a belief in the self's inadequacy. The paranoid mode then consists of strategies which forestall or ward off an impending negative affect experience of humiliation by negating this belief that the self is wrong and asserting the belief that the self is being wronged by others. The theory provides generative mechanisms for explaining the expression of a delusional system by a paranoid person, the chronic distress felt by paranoid persons, and for the sudden and extreme displays of fear and anger in interactions with other people.

We plan to extend the theory in two ways:

- (1) To cover other paranoid phenomena, such as delusions of grandeur, the transformation of counter-evidence to evidence supportive of delusions, and retrospective misinterpretation of input expressions.
- (2) To explain the genesis of paranoid patterns of thought; e.g., the manner in which: (a) normal strategies for dealing with the shame-humiliation affect are ineffective and paranoid strategies develop, (b) strategies are selected as being appropriate and are reinforced when they are successful, and (c) persecutory delusional systems develop and expand to include much of the paranoid personality's cognitive processing.

B. Expand the background theory and model of the motivation of cognitive processes. The more important characteristics for the model to have are:

- (1) The top-level processes of the model should be purposive intentional processes guided by the affect system, rather than a question-answer loop or facsimile.
- (2) Every action that the model performs should be motivated by an intention. These intentions may be explicit in the case of goals, or implicit in the case of an action appropriate to a situation but with no explicit end state represented.
- (3) Each belief and intention should have an associated measure of its significance to the entity. The criteria for measuring the significance are based in the affect system.
- (4) The model should have a number of coping mechanisms for avoiding or coping with distressing situations. These mechanisms can be reinforced or discarded as they are proved to be more or less useful to the model.
- (5) The model should be able to change over time to show the development of psycho-pathologies. The most direct form of change is to the measures of significance attached to the model's beliefs and intentions.

C. Implement the theories in a simulation model. We hope to construct a model in such a way that the theories of motivation and paranoia can be represented explicitly, and therefore be open to inspection and modification. In addition, since we model paranoid behavior as expressed through linguistic actions, we hope to develop adequate natural language understanding programs for recognition and response in dialog situations.

D. Develop further techniques and methods for simulating cognitive processes. Specifically, we plan to explore human communication through natural language in dialog situations and human natural language interfaces with computers. Also, we will extend our previous attempts at finding stronger and more sophisticated tests for validation studies. Our

results should be applicable to other simulations of human mental activities.

Medical relevance.

The simulation model of paranoid processes that we are implementing has implications for the understanding, treatment, and management of paranoid disorders.

The shame-humiliation theory and its model suggest that paranoid phenomena be viewed as a consequence of intentionalistic information-processing strategies which attempt to avoid or minimize the distressing experience of humiliation. In trying to understand what is going on in a paranoid patient at a symbol-processing level, this perspective directs clinicians to look for humiliating and shame-engendering situations in the patient's experience. These may consist of a single, encompassing humiliating situation such as a demeaning job, or a series of esteem-damaging defeats such as repeated failures in disappointing love affairs.

Since activation of intense shame is posited to be the core process in paranoid disorders, implications for treatment involve trying to modify this central mechanism in some way. One method is to change the patient's distressing belief in his own inadequacy by exploring topics involving shame, esteem, and self-censure. Another is to desensitize the paranoid patient to shame experiences through behavior therapy involving a graded hierarchy of imagined distress situations and countering procedures. These treatment procedures may be deduced directly from mechanisms in the model, and the theory used to predict the outcomes of such procedures.

For management of the disorder, the model predicts that removal from situational humiliation, as in hospitalization, allows for repair from breakdowns occurring under repeated activations of shame-humiliation beliefs. Also, if the patient returns to unchanged situational humiliation, as in a distressing home life, he risks a relapse.

Current Status.

PARRY2 was completed a year ago and has been available for interviewing and validation tests on the SUMEX system for the past year. We are now in the process of writing a new version, PARRY3, incorporating the theoretical constructs presented in this report. The new version (which is being completely rewritten) contains mechanisms implementing the characteristics of models mentioned above with the exception of the ontogenesis of psycho-pathologies. In addition, it contains a new language recognizer capable of combining pattern-matching rules with parsing techniques, and explicit rules for recognizing and interpreting elliptical expressions in dialogs.

Current publications.

[1] Colby, K.M. Artificial Paranoia, Pergamon Press, New York, 1975.

- [2] Faught, W.S. Affect as Motivation for Cognitive and Conative Processes, IJCAI Proceedings, September, 1975.
- [3] Colby, K.M., Hilf, F.D., Wittner, W.K., Parkison, R.C. and Faught, W.S., A Note on Estimating Improvement in a Computer Simulation of Paranoid Processes. UCLA Department of Psychiatry, Memo ALHMF-2, April 1975.
- [4] Parkison, R.C., Colby, K.M., and Faught, W.S., Conversational Language Comprehension Using Integrated Pattern-Matching and Parsing. UCLA Department of Psychiatry, Memo ALHMF-5, April 1976.
- [5] Colby, K.M. Clinical Implications of a Simulation Model of Paranoid Processes. Archives of General Psychiatry, 1976.
- [6] Faught, W.S., Colby, K.M., and Parkison, R.C., Inferences, Affects, and Intentions in a Model of Paranoia. Cognitive Psychology, 1976.

Funding Status.

Grant NIMH , 2 years, \$67,000 this year.

Interactions with the SUMEX-AIM resource.

The SUMEX-AIM resource and its associated network connections make possible the merging of artificial intelligence techniques and technology in psychiatry and the resources of a west-coast center for psychiatric studies, the Neuro Psychiatric Institute (NPI) at UCLA. Access to SUMEX from the NPI has brought a new source of questions and viewpoints to research in mental disorders, as well as an opportunity for the model-builders to interact with clinicians in elaborating details of paranoid phenomena. In addition, the current simulation model is being explored for use as a training device for medical students and residents in the Department of Psychiatry.

Critique of resource services.

The resource itself has provided excellent facilities and service for our research needs. We have had almost no trouble developing simulations due to the SUMEX system itself, and the cooperation of the SUMEX staff has been excellent.

In spite of this, problems have arisen with our use of the resource, due almost entirely to the network aspect of our access. (We connect to SUMEX through both the ARPA net and TYMNET.) We see the problems of network use of the SUMEX-AIM facility as falling into four broad categories:

- (a) the keyhole effect. The slow terminal rates typical of a network connection (due to phone lines or local computer delay) force the user to peer at his files and communicate with the computer through

a small data channel. Additionally, the network user may not have direct access to a high-speed printing device for listing the day's (or even the week's) work.

- (b) the dropped connection. Typically, the network and/or local connection of the user to the net can drop, leaving the user's job in a dormant (non-running) state, forcing the user to reconnect (and perform the typically elaborate reconnect procedure) before his job will run again.
- (c) slower interactive computer response due to network and local hardware, and subsequent greater amount of time necessary to get any work done.
- (d) the difficulty of lobbying for system changes and/or additions from a distance. We are made acutely aware of this fact whenever notices are given over the system for classes or seminars explaining new system features.

Policies and programs that might be useful:

- (a) more system programs designed with the network user in mind, such as status programs to report the status of detached or batch-run jobs, and cleaner detach and attach programs for reestablishing dropped connections.
- (b) a service-level advantage to non-local users to put them on a more equal footing with local users.
- (c) a special effort to elicit and implement improvements for network access.

In general, we have found the system reliable, and the staff courteous and helpful.

IV.A.2.d LANGUAGE ACQUISITION MODELING PROJECT

Language Acquisition Modeling (ACT)

Dr. John Anderson
(Univ. of Michigan)

(Grant NIMH \$20,000 this year)

I. Summary of Research Program

A. Technical goals:

To develop a production system that will serve as an interpreter of the active portion of an associative network. To model a range of cognitive tasks including memory tasks, inferential reasoning, language processing, and problem solving. To develop an induction system capable of acquiring cognitive procedures with a special emphasis on language acquisition.

B. Medical relevance and collaboration:

1. The ACT model is a general model of cognition. It provides a useful model of the development of and performance of the sorts of decision making that occur in medicine.

2. The ACT model also represents basic work in AI. It is in part an attempt to develop a self-organizing intelligent system.. As such it is relevant to the goal of development of intelligent artificial aids in medicine.

C. Progress and Accomplishments:

ACT provides a uniform set of theoretical mechanisms to model such aspects of human cognition as memory, inferential processes, language processing, and problem solving. ACT's knowledge base consists of two components, a propositional component and a procedural component. The propositional component is provided by an associative network encoding a set of facts known about the world. This provides the system's semantic memory. The procedural component consists of a set of productions which operate on the associative network. ACT's production system is considerably different than many of the other currently available systems (e.g., Newell's PSG). These differences have been introduced in order to create a system that will operate on an associative network and in order to accurately model certain aspects of human cognition.

A small portion of the semantic network is active at any point in time. Productions can only inspect that portion of the network which is active at the particular time. This restriction to the active portion of

the network provides a means to focus the ACT system in a large data base of facts. Activation can spread down network paths from active nodes to activate new nodes and links. To prevent activation from growing continuously there is a dampening process which periodically deactivates all but a select few nodes. The condition of a production specifies that certain features be true of the active portion of the network. The action of a production specifies that certain changes be made to the network. Each production can be conceived of as an independent "demon". Its purpose is to see if the network configuration specified in its condition is satisfied in the active portion. If it is, the production will execute and cause changes to memory. In so doing it can allow or disallow other productions which are looking for their conditions to be satisfied. Both the spread of activation and the selection of productions are parallel processes whose rates are controlled by "strengths" of network links and individual productions. An important aspect of this parallelism is that it is possible for multiple productions to be applied in a cycle through the set of productions. Much of the early work on the ACT system was focused on developing computational devices to reflect the operation of parallel, strength-controlled processes and working out the logic for creating functioning systems in such a computational medium.

We have successfully implemented a number of small-scale systems that model various psychological tasks in the domain of memory, language processing, and inferential reasoning. A larger scale effort is underway to model the language processing mechanisms of a young child. This includes implementation of a productions system to analyze linguistic input, make inferences, ask and answer questions, etc. Also a great deal of effort is being given to developing learning mechanisms that will acquire and organize the productions for this language processing. This learning program attempts to acquire procedures from examples of the computations desired of the procedures. For instance, the program learns to comprehend and generate sentences by being given sentences and picture representations of the meaning of the sentences (actually hand encodings of the pictures). Although this effort is focused on induction of linguistic procedures, the hope is to develop a general model of induction of cognitive procedures and not to place any language-specificity into the induction procedures.

D. Current list of project publications:

- [1] Anderson, J. R. Induction of augmented transition networks. Cognitive Science, 1976, in press.
- [2] Anderson, J. R. Language, Memory, and Thought. Lawrence Erlbaum Assoc., 1976, in press.
- [3] Anderson, J. R., Klein, P., and Lewis, C. Language processing by production systems. To appear in P. Carpenter and M. Just (Eds.) Cognitive Processes in Comprehension, Lawrence Erlbaum Assoc.

E. Funding Status:

The research is currently being funded by a grant from NIMH for computer simulation of language acquisition. The level of funding for the year beginning May 1, 1976 has yet to be determined. It was \$20,000 for the past year.

II. Interactions with SUMEX-AIM Resource.

Our period with the project has been too short to develop any significant interactions. The ACT program is currently being made a system which will be available to members of the SUMEX-AIM community.

IV.A.2.e MEDICAL INFORMATION SYSTEMS LABORATORY

MISL Project

Dr. B. McCormick and M. Goldberg, M.D.
 (Univ. of Illinois at Chicago Circle)

(Grant HEW MB-00114-02, 3 years, \$248,793 this year)

I) Summary of research program

A) Technical goals

The major goals of the Medical Information Systems Laboratory fall into three broad categories, described briefly as follows:

- 1) Construction of a database in ophthalmology. This will provide a trial setting for clinical decision-making research. Four major activities are involved: implementation of a clinical data network; design and on-going development of an Eye Outpatient Index; computer systems development; and installation of a glaucoma clinic satellite.
- 2) Network-compatible database design. This will provide cost-effective distributed data management for clinical records. Current projects include: design of an intelligent coupler for the database system; continuing design of various levels of database software (a relational algebraic language -- RAIN, disk controller, database skeleton); large database / database software compatibility; design of a separate database computer; design of database network software.
- 3) Clinical decision support
 - a. Construction of a consultation system for use in the diagnosis and treatment of retinal/choroidal diseases. The immediate goal is the development of a system for giving advice about four diseases prevalent at the University of Illinois Eye and Ear Infirmary: histoplasmosis, central serous retinopathy, diabetic retinopathy, and sickle cell disease. Besides actual construction of a system, the project is concerned with theory of diagnosis and knowledge acquisition methodology.
 - b. Interface between a pattern recognition system for Structured Analysis of the Retina (STARE) and the diagnostic model used by the retinal/choroidal consultation system.
 - c. Initial development of a causal model for motility.
 - d. Further inter-institutional communications in disease model theory and development.

B) Medical relevance and collaboration

We have chosen to explore inferential relationships between analytic clinical data and the natural history of glaucoma and selected retinal/choroidal diseases both in treated and untreated form. These investigations are intended to provide much more than a simple excursion of computer technology into ophthalmology. They address clinical problems of national interest, as indicated in the Report of the National Eye Advisory Council Vision Research Program Planning Committee (DHEW publication No. NIH-75-644).

Glaucoma, one of the major causes of blindness in the United States today, is difficult to diagnose in its early stages. Some recent evidence indicates that enlargement of the optic nerve cups may be the first sign of glaucoma's damage to the eye. One of the goals of the present project is the application of a newly developed technique for quantitative analysis of the optic nervehead. The technique is sufficiently simple to permit wide-spread adoption. If this technique is successful in identifying very early glaucomatous disk changes, it should permit institution of therapy at a very early stage, and thereby prevent serious glaucomatous damage from being done to the eye.

Diabetic retinopathy is another principal cause of blindness. Very little is known about its pathophysiology, and there are many gaps in our knowledge of its natural course. The present study is designed to elicit new information about this disease, using a series of new diagnostic tools which have been developed as part of a system of computerized retinal image analysis. The need here is great, because at present there is no proven satisfactory treatment.

Sickle hemoglobinopathy can cause ocular changes that lead to loss of vision and even total blindness. Little is known about the natural history of this problem, particularly during its early stages. The present project is ideally suited to assist in this study; the nation's first Sickle Cell Clinic has been established at the Illinois Eye and Ear Infirmary -- the site of our Ophthalmic Database System.

At present there is great demand in the United States for improved efficiency in the delivery of medical care. Two ways that this can be accomplished are: 1) by increased use of paramedical personnel to perform jobs currently being performed by physicians, and 2) by use of automated equipment to perform tasks previously performed by the physician. In the current project we utilize both these methods in the screening of new ophthalmic patients at the Illinois Eye and Ear Infirmary. If we can show that these methods are not only feasible, but also improve the efficiency and reliability of patient care, then a major contribution to ophthalmic care for patients in large ambulatory care centers around the country will have been made.

Modeling of clinical decision-making is best carried out in intimate association with an extensive referral clinic where a sufficient patient population can be accumulated to provide an adequate biostatistical sample. In the prescribed setting, the Illinois Eye and Ear Infirmary, the Medical Information Systems Laboratory has access to:

- clinical expertise provided by a house staff of 45;

- 28 residents, all of whom are required to undertake some research work as a requirement of their appointment, are available to explore latent contingencies of the database;
- an indigenous, relatively stable population (25% white, 75% black and other minorities) of a medically underserved portion of the inner city of Chicago; the clinic provides ophthalmic services to 50,000 patients per year.

Commonality of the diseases being studied assists construction of an adequate biostatistical sample. Roughly 2% of the general population exhibit symptoms of diseases treated at each of three specialty clinics (Glaucoma, Retina, and Motility), or allowing for multiple presenting of symptoms, approximately 5% by population.

Besides a strong clinical research orientation, the Medical Information Systems Laboratory brings to the study of disease a history of successful engineering-medical collaboration. MISL's sister project, "Image Processing in Clinical Ophthalmology," lists the development of the digital television ophthalmoscope as one of its achievements. This device will be a major source of clinical data for our Ophthalmic Database.

C) Progress and accomplishments (of the Clinical Decision Support activity only)

Interaction has continued over SUMEX-AIM with the authors of the Weiss/Kulikowski glaucoma modeling program. We are now entering cases into the glaucoma system at the rate of approximately 5 per week.

Work on the consultation system for retinal/choroidal diseases has progressed along two fronts. While interviewing an expert diagnostician, in order to build a knowledge base for the four diseases mentioned above, we have been piecing together a theory of diagnosis for ocular fundus diseases. We have attempted to incorporate pieces of the expert's knowledge in a "fuzzy" diagnostic model, based partly on multiple-cue probability theory, partly on fuzzy set and confirmation theory. The framework of the model is a hierarchy of disease categories, each with a significance tempered by functions built into the system. Our efforts have centered on the acquisition of categories for histoplasmosis, central serous retinopathy, and diabetic retinopathy, but should soon also include sickle cell disease.

Our experience in interviewing experts has pointed the way to a knowledge acquisition methodology that is compatible with our thoughts on diagnostic reasoning. Specifically we plan to store, for each disease category, a representation for the contexts (or frames) in which the disease's attributes apply. We conceive of a disease category as a "sphere" (embodying a structural model of the disease) in a hyperspace defined by dimensions on attributes. The significance of (or "belief in") the model of disease is modified by interactions between attributes. During acquisition, and after contexts have been defined, we can simulate clinical situations for the benefit of our expert, who indicates his level of belief in the disease model in the given situation. This we plan to do

with the help of plasma panels, for graphical presentations of the relevant contexts comprising each situation. This approach is especially convenient for specifying typical "default" situations, and for modeling the time course of disease (in terms of modifications on attributes).

Presently, while we continue interviews with our expert, we are formalizing our diagnostic model and expect shortly to finish an in-depth report.

D) Current list of project publications

- [1] Chang Shi-Kuo, O'Brien M., Read J., Borovec R., Cheng W. H., Ke J. S. (1976) Design considerations of a database system in a clinical network environment. Accepted for 1976 National Computer Conference.
- [2] Chang S. K. (1975) Preliminary report on the implementation of a relational data base management system with structurally decomposed relations. MISL internal report M.D.C. 1.1.3.
- [3] Chang S. K. and McCormick B. H. (1975) An intelligent coupler for distributed database systems. MISL internal report M.D.C. 1.1.7.
- [4] Malone J. (1975) User's guide to uniclass cover synthesis. MISL report M.D.C. 4.4.1.
- [5] Malone J. (1975) Addendum to AQVAL/1 (AQ7), part 1: User's guide and program description. MISL report M.D.C. 4.4.1.
- [6] Manacher G. K. (1975) On the feasibility of implementing a large relational data base with optimal performance on a minicomputer. Proceedings of the International Conference on very large data bases, Framingham, Mass.
- [7] McCormick B. H. and Wilensky J. (1975) Clinical knowledge acquisition: design of a relational data base in ophthalmology. To be published in Proceedings Second Annual Medical Informations Systems Conference, Urbana, Ill.
- [8] McCormick B. H., Goldberg M. F., and Read J. S. (1974) Clinical decision-making: design of a data base in ophthalmology. Proceedings First Annual Medical Information Systems Conference, Urbana Ill.
- [9] Michalski, Ryszard S. (1975) On the selection of representative samples from large relational tables for inductive inference. MISL internal report M.D.C. 1.1.9.
- [10] Murata T. (1976) Equations governing the behavior of E-nets. MISL internal report.
- [11] Murata T. (1975) State equation, controllability, and maximal matchings of petri nets. MISL report M.D.C. 1.1.10.
- [12] Murata T. and Church R. W. (1975) Analysis of marked graphs and petri nets by matrix equations. MISL report M.D.C. 1.1.8.

- [13] Vere S. A. (1975) Induction of concepts in the predicate calculus. Proceedings of the Fourth International Joint Conference on Artificial Intelligence vol 1., Tbilisi, U.S.S.R.
- [14] Vere S. A. (1975) Relational productions systems. MISL internal report M.D.C. 1.1.5.

E) Funding status

Year 02 -- 6/30/75 - 6/29/76 : \$248,793.

Year 03 -- 6/30/76 - 6/29/77 : \$228,000.

II) Interactions with the SUMEX-AIM Resource

A) & B) Collaboration, cross-fertilization

Most of our interaction of late has involved the Glaucoma Network fostered by the Rutgers Computers in Biomedicine group. This network has made it especially convenient for our expert in glaucoma, Dr. Jacob Wilensky, to maintain close contact with investigators around the country.

In addition, monitoring of SUMEX-AIM system messages has helped us keep abreast of developments in other projects. We have come to rely on this facility as a vital source of up-to-date information.

C) Critique of resource services

In our view SUMEX-AIM services are excellent. We have been very pleased with the prompt and personal attention given to our requests by the resource staff.

IV.A.2.f RUTGERS COMPUTERS IN BIOMEDICINE

Rutgers Research Resource
Computers in Biomedicine

Dr. Saul Amarel
Rutgers University
New Brunswick, New Jersey

(Grant NIH RR-00643-05, 3 years, \$314,880 this year)

I. PROJECT GOALS AND APPROACHES

The fundamental objective of the Rutgers Resource is to develop a computer based framework for significant research in the biomedical sciences and for the application of research results to the solution of important problems in health care. The focal concept is to introduce advanced methods of computer science - particularly in artificial intelligence - into specific areas of biomedical inquiry. The computer is used as an integral part of the inquiry process, both for the development and organization of knowledge in a domain and for its utilization in problem solving and in processes of experimentation and theory formation.

The Resource community includes 46 researchers - 26 members, 8 associates and 12 collaborators. Members are mainly located at Rutgers. Collaborators are located in several distant sites and they interact - via SUMEX-AIM - with Resource members on a variety of projects, ranging from system design/improvement to clinical data gathering and system testing. At present, collaborators are located at the Mt. Sinai School of Medicine, N.Y.; Washington University School of Medicine, St. Louis, Mo.; Johns Hopkins Medical Center, Baltimore, Md.; Illinois Eye and Ear Infirmary, Chicago, Ill.; College of Medicine and Dentistry of New Jersey (CMDNJ); and the University of Miami.

Research in the Rutgers Resource is oriented to "discipline-oriented" projects in medicine and psychology, and to "core" projects in computer science, that are closely coupled with the "discipline-oriented" studies. Work in the Resource is organized in three AREAS OF STUDY; in each area there are several projects. The areas of study and the senior investigators in each of them are:

- (1) Medical Modeling and Decision Making (C. Kulikowski, A. Safir).
- (2) Modeling Belief Systems (C. F. Schmidt, N. S. Sridharan).
- (3) Representations, Modeling and Hypothesis Formation in AI (S. Amarel).

In addition the Rutgers Resource is sponsoring an Annual National AIM Workshop, whose main objective is to strengthen interactions between AIM activities, to disseminate research methodologies and results, and to stimulate collaborations and imaginative resource sharing within the

framework of AIM. The first AIM Workshop was held at Rutgers on June 14 to 17, 1975. The Second Workshop is scheduled for June 1 to 4, 1976.

II. AREAS OF STUDY; SUMMARY OF GOALS AND ACTIVITIES

(1) Medical Modeling and Decision-Making

Present projects include:

- (i) Development and clinical testing of the glaucoma consultation program based on a causal-associational network (CASNET) model - as a collaborative project of the ophthalmological network, which has grown in the last year to include: Mt. Sinai School of Medicine, Washington University, Johns Hopkins University, University of Illinois at Chicago, and the University of Miami.
- (ii) Investigation of descriptive models of diseases based on a general semantic network representation, with associated strategies of diagnosis, prognosis, and therapy. These models subsume a variety of sub-models useful in general consultation. A particular emphasis is placed on the analysis of the time course of disease, and inter-relationships between various disease sub-processes.
- (iii) Development of a data base for clinical research associated with the consultation programs. The results of the data analyses to be used selectively in updating and improving the models of diseases.
- (iv) Investigation of various techniques for acquiring knowledge from clinical experts. Incorporation of alternative expert opinions within a model.
- (v) In collaboration with the Mt. Sinai Rutgers Health Care Computer Laboratory, we are developing models for refraction and neuro-ophthalmology.

The following is a summary of accomplishments in this area.

- (a) The ophthalmological network(ONET) is actively underway, with consultation programs available through SUMEX-AIM to the five collaborating institutions.
- (b) The consultation system has been perfected by adding many details of diagnosis, prognosis, and therapy; new observations and decision criteria have been specified as the result of suggestions by the ONET collaborators.
- (c) A data base for storing cases and providing a chronological model based interpretation has been created.

- (d) A set of programs to analyze the case histories has been developed. They are currently being used in collaborative clinical studies by ONET members. Selected results are to be incorporated into the glaucoma model.
- (e) A semantic network based model for glaucoma has been implemented with expanded capabilities for explanation and a greater facility for being updated.

The progress in extending the collaborative activities of the ophthalmological network has been made possible by the facilities and support provided by SUMEX-AIM. The semantic network model is being developed on INTERLISP at SUMEX-AIM. Other program development activities are evenly distributed between SUMEX-AIM and the Rutgers-10.

(2) Modeling Belief Systems

The overall goal of this project is to develop a computer based psychological model of how persons reason about the causes of human action. The common-sense notion of social causation which is used to understand intentional actions has served as the focus of this effort. The construction of a system (called BELIEVER) to assist in the description of the psychological theory and to facilitate the testing of the theory is a principal goal of this project and to date we have accomplished the following:

- (a) A working system has been developed that accepts the descriptions of Templates, Relations and their consistency conditions which are concepts developed in the Meta Description System (MDS) framework.
- (b) We have described PLANS, ACTS, PERSONS and SUMMARIES as templates; we have defined the consistency conditions associated with the relations among these.
- (c) Developed procedures in FUZZY for Instantiating Act Descriptions and calculating presuppositions needed to form coherent interpretations.
- (d) We have provided for an easy-to-understand prompting scheme based on the concept of templates and consistency conditions. The prompting proceeds using the template descriptions and attempts to fill in missing information by implication following based on the consistency conditions.
- (e) Developed a framework for submitting and analyzing the semantics of experimental evidence when the subject responses are solely unstructured natural language text.

Our goals for the immediate future are:

- (a) To continue to gather experimental data and follow the analysis to suggest hypothesis in the Believer Theory.

- (b) Develop strategy Plan Recognition based on a Theory of Motivation and Consistency postulates of a Persons beliefs and knowledge.
- (c) Continue to investigate the innovative uses of our descriptive methodology in empirical procedures.

The development of MDS - which provides a framework for designing the BELIEVER system - was carried out at SUMEX-AIM until October 1975; subsequently, most of the work on MDS was shifted to ISID. While early versions of BELIEVER were developed at SUMEX-AIM, the last year saw a shift in computing on this project from SUMEX-AIM to the Rutgers-10 - since program development is being done on the FUZZY system which runs on the Rutgers computer.

(3) Representations, Modeling and Hypothesis Formation in Artificial Intelligence

A major part of our effort in this area is oriented to collaborations with investigators in other Resource projects - involving applications of AI ideas and programs and also identification and initial exploration of new significant AI problems. The collaboration in the BELIEVER area has lead to a close integration of the basic AI oriented work (N. S. Sridharan) and the discipline oriented work (C. Schmidt) in the area. This work is reported under 2 above.

"Core" projects in the present area include:

- (i) Development and applications of FUZZY (R. LeFaivre). The FUZZY system was transferred from UNIVAC 1110 LISP to UCI LISP on the Rutgers-10. A number of improvements were made, both to the UCI LISP system and to the FUZZY language to make it both more efficient and more powerful. These changes include a new prettyprint package for UCI LISP and functions for computing differences of associative nets and creating multiple contexts in FUZZY. FUZZY is currently being used in the initial implementation phase of the BELIEVER system. Applications to reasoning in medical diagnosis are being explored.
- (ii) Applications of grammatical inference schemes to automatic adjustment of medical models on the basis of clinical data (A. Walker).
 - (a) An algorithm was found to eliminate loops from stochastic causal models.
 - (b) A grammar model for the flow of control in programs was formulated.
 - (c) A technique for progressively bounding a search space of stochastic grammars, in terms of grammars already found, was studied theoretically and tested in practice.
 - (d) The impact of this work on the automated construction of production-rule based AI systems is now under study.

Computing in this project is done on the Rutgers-10.

- (iii) Development of a grammatical inference system using a "developmental paradigm" (W. Fabens). This is a hypothesis formation system which attempts to change a given context free grammar so as to accommodate new sentences that cannot be derived from the given grammar. The system includes (a) a relaxation parser - which comes as close as it can to an interpretation of a given "deviant sentence", (b) a rule hypothesizer which uses such an interpretation to propose changes to the current grammar, (c) a rule coalescer which summarizes with as little loss or gain in generality as possible the newly hypothesized grammar. We have developed programs in all of these areas and are currently composing them into a single system. Computing in this project is done on the Rutgers-10.
- (iv) Development and study of systems for theory formation in programming tasks (S. Amarel). A system is being developed for acquiring knowledge about a program formation task. The system involves the generation, management and evaluation of programs at various stages of specification. In this project, major emphasis is given to problems of representation and to the effect of shifts between representations. Program development is being done on the Rutgers-10.

While the first project discussed in this area is focusing on the development of AI languages that can provide a stronger basis for system development in the Resource, the last three projects are focusing on different AI approaches to hypothesis (theory) formation - an area which is essential to the automatic acquisition and improvement of a knowledge base from experimental data.

III. AIM WORKSHOP

The first annual AIM Workshop was held June 14-17, 1975. The first day was devoted to a General Session to provide an overview of current AIM activities and a broad forum for discussion. The following three days were devoted to discussions in depth of AIM designs, and to demonstrations of current systems. Several AIM systems were effectively demonstrated on SUMEX during the Workshop.

The second annual AIM Workshop will be held June 1-4, 1976. The entire four days will be devoted to lectures and panel discussions on current projects in the AIM community, and on problems of knowledge representation, reasoning, and AI system design. Papers on language, speech, vision, education and problem solving will summarize recent AI approaches, while the role of biomathematical modeling and inference methods will be the focus of summary papers and panel discussions. Tutorials on languages and systems available to the AIM community will also be presented. The Workshop will conclude with a panel on the dissemination of scientific information and computer networking.

The SUMEX-AIM system is essential for the Workshop. Many of the AIM programs will be running on SUMEX-AIM and accessed via TYMNET or ARPANET

from Rutgers. The message facilities of SUMEX-AIM are most useful for planning, communicating and setting up the information pool for the AIM Workshop.

IV. INTERACTIONS WITH THE SUMEX-AIM RESOURCE

During the past year we have continued to use the SUMEX-AIM resource for program development and testing, for communications between collaborators distributed in different parts of the country and for preparation and running of the AIM Workshop.

Computing in the Rutgers Research Resource is distributed between SUMEX-AIM and the Rutgers-10. The relative utilization of SUMEX-AIM by "local" Rutgers users has decreased this year. The utilization of SUMEX-AIM by our "remote" medical collaborators has been growing. The total amount of computing at SUMEX by our Resource users and by our collaborators has decreased relative to the previous year. One of the reasons for this was the overloaded condition of SUMEX-AIM. Another important related problem was the relatively poor quality of communication facilities available to us via TYMNET.

In order to provide a more reliable and convenient network environment for our investigators and their collaborators and also for the AIM Workshop, we have proceeded this year with the implementation of several enhancements to the Rutgers-10. These enhancements were planned in consultation with AIM management, with the intention of bringing to the AIM network complementary facilities and added capacity. Two stages of enhancement have been completed this year: (a) core memory and fixed head disk were augmented and the TOPS 6.02 operating system was installed; (b) a TYMCOM communications unit was installed, making the Rutgers-10 accessible via TYMNET - in time for support of the second AIM workshop.

The SUMEX-AIM facility played a key role this year in consolidating our network of collaborators in ophthalmology (ONET) and providing the support needed for establishing a productive collaboration among the ONET investigators.

The SUMEX staff have continued to function as models of excellent cooperation. They have been very helpful and responsive in sharing information and keeping us aware of developments, problems, new ideas, etc. SUMEX continues to be a good forum for communicating, linking and talking with various investigators in our Resource as well as with others in the AIM community.

The AIM Workshops rely heavily on SUMEX-AIM. In the first Workshop, the demo sessions and the hands-on activities with remote systems were found to be very effective in disseminating AIM methods and techniques to a broad group of participants. The significant role played in these demos by the SUMEX staff, and by the SUMEX resource, cannot be overstated. In our planning for the second AIM Workshop we are also placing strong emphasis on on-line activities, more so considering the broader class of participants who will meet this year.

SUMEX-AIM has been most useful in communicating, planning and helping to set up the information pool for this AIM Workshop.

In conclusion, the SUMEX-AIM facility is continuing to be an essential part of our research environment. In view of our AIM Workshop activities and the related enhancement of the Rutgers-10, we are moving to a point where the interactions between the Rutgers project and SUMEX-AIM are increasing in scope - as Rutgers is gradually adding to its "user" role a "server" role for the national AIM project.

IV. LIST OF PROJECT PUBLICATIONS

- [1] Amarel, S., and Kulikowski, C. (1972) "Medical Decision Making and Computer modeling, Proc. of 5th International Conference on Systems Science, Honolulu, January 1972.
- [2] Amarel, S. (1974) "Computer-Based Modeling and Interpretation in Medicine and Psychology: The Rutgers Research Resource", Proc. on "Conference on the Computer as a Research Tool in the Life Sciences", June 1974, Aspen, by FASEB; also appears as Computers in Biomedicine TR-29. June 1974, Rutgers University.
- [3] Amarel, S. (1974) "Inference of Programs from Sample Computations", Proc. of NATO Advanced Study Institute on Computer Oriented Learning Processes, 1974, Bonas, France.
- [4] Bruce, B., (1972) "A Model for Temporal Reference and its Application in a Question Answering Program", in "Artificial Intelligence", Vol. 3, Spring 1972.
- [5] Bruce, B. (1973) "A Logic for Unknown Outcomes", Notre Dame Journal of Formal Logic; also appears as Computers in Biomedicine, TM-35, Nov. 1973, Rutgers University.
- [6] Bruce, B. (1973) "Case Structure Systems", Proc. 3rd International Joint Conference on Artificial Intelligence (IFCAI), August 1973.
- [7] Bruce, B. (1975) "Belief Systems and Language Understanding", Current Trends in the Language Sciences, Sedelow and Sedelow (eds.) Houton, in press.
- [8] Chokhani, S. and Kulikowski, C.A. (1973) "Process Control Model for the Regulation of Intraocular Pressure and Glaucoma", Proc. IEEE Systems, Man & Cybernetics Conf., Boston, November 1973.
- [9] Chokhani, S. (1975) "On the Interpretation of Biomathematical Models Within a Class of Decision-Making Procedures", Ph.D. Thesis, Rutgers University; also Computers in Biomedicine TR-43, May, 1973.
- [10] Fabens, W. (1972) "PEDAGLOT. A Teaching Learning System for Programming Language", Proc. ACM Sigplan Symposium on Pedagogic Languages, January 1972.

- [11] Kulikowski, C.A. and Weiss, S. (1972) "Strategies for Data Base Utilization in Sequential Pattern Recognition", Proc. IEEE Conf. on Decision and Control, Symp. on Adaptive Processes, December 1972.
- [12] Kulikowski, C.A. and Weiss, S. (1973) "An Interactive Facility for the Inferential Modeling of Disease", Proc. 7th Annual Princeton Conf. on Information Sciences and Systems, March 1973.
- [13] Kulikowski, C.A. (1973) "Theory Formation in Medicine: A Network Structure for Inference", Proc. International Conference on Systems Science, January 1973.
- [14] Kulikowski, C.A., Weiss, S., and Safir, A. (1973) "Glaucoma Diagnosis and Therapy by Computer", Proc. Annual Meeting of the Association for Research in Vision and Ophthalmology, May 1973.
- [15] Kulikowski, C.A. (1973), "Medical Decision-Making and the Modeling of Disease", Proc. First Intern. Conf. on Pattern Recognition, October, 1973.
- [16] Kulikowski, C.A. (1974) "Computer-Based Medical Consultation"-- A Representation of Treatment Strategies", Proc. Hawaii International Conf. on Systems Science, Jan. 1974.
- [17] Kulikowski, C.A. (1974) "A System for Computer-Based Medical Consultation", Proc. National Computer Conference, Chicago, May 1974.
- [18] Kulikowski, C.A. and Safir, A. (1975) "Computer-Based Systems Vision Care", Proceedings IEEE Intercon, April 1975.
- [19] Kulikowski, C. and Trigoboff, M. (1975a) "A Multiple Hypothesis Selection System for Medical Decision-Making", Proc. 8th Hawaii International Conference on Systems Sciences, January 1975.
- [20] Kulikowski, C. & N.S. Sridharan, "Report on the First Annual AIM Workshop on Artificial Intelligence in Medicine. Sigart Newsletter No. 55, December 1975.
- [21] Kulikowski, C., "Computer-Based Consultation Systems as a Teaching Tool in Higher Education, 3rd Annual N.J. Conference on the use of Computers in Higher Education, March 1976.
- [22] Kulikowski, C., Weiss S., Safir, A. et al "Glaucoma Diagnosis & Therapy by Computer: A Collaborative Network Approach" Proc. of ARVO, April 1976.
- [23] Kulikowski, C., Weiss, S., Trigoboff, M. Safir, A., "Clinical Consultation and the Representation of Disease Processes", Some AI Approaches, AISB Conference, Edinburgh, July 1976.
- [24] LeFaivre, R., "Procedural Representation in a Fuzzy Problem-Solving System", Proc. Natl. Computer Conf., New York, June 1966

- [25] LeFaivre, R. and Walker, A. "Rutgers Research Resource on Computers in Biomedicine, H", Sigart Newsletter No. 54, October 1975.
- [26] Mauriello, D. (1974) "Simulation of Interaction Between Populations in Freshwater Phytoplankton", Ph.D. Thesis, Rutgers University, 1974.
- [27] Schmidt, C. (1972) "A comparison of source unidimensional, multidimensional and set theoretic models for the prediction of judgements of trail implication", Proc. Eastern Psych. Assoc. Meeting, Boston, April 1972.
- [28] Schmidt, C.F. and D'Addamio, J. (1973) "A Model of the Common Sense Theory of Intension and Personal Causation", Proc. of the 3rd IJCAI, August 1973.
- [29] Schmidt, C.F. and Sedlak, A. (1973) "An Understanding of Social Episodes", Proc. of Symposium on Social Understanding in Children and Adults: Perspectives in Social Cognition, American Psych. Assoc. Convention, Montreal, August 1973.
- [30] Schmidt, C.F., Sridharan, N.S., & Goodson, J.L. Recognizing plans and summarizing actions. Proceedings of the Artificial Intelligence and Simulation of Behavior Conference, University of Edinburgh, Scotland. July 1976.
- [31] Schmidt, C. Understanding human action: Recognizing the plans and motives of other persons. In (eds. J. Carroll and J. Payne) Cognition and Social Behavior. Potomac, Maryland:
- [32] Schmidt, C.F. (1975) "Understanding Human Action", Proc. Theoretical Issues in Natural Language Processing: An Interdisciplinary Workshop in Computational Linguistics, Psychology, Artificial Intelligence, Cambridge, Mass., June 1975. Also appears as Computers in Biomedicine, TM-47, June 1975, Rutgers University.
- [33] Schmidt, C.F. (1975) "Understanding Human Action: Recognizing the Motives", Cognition and Social Behavior, J.S. Carroll and J. Payne (eds.), New York: Lawrence Earlbaum Associates, in press. Also appears as Computers in Biomedicine, TR-45, June 1975, Rutgers University.
- [34] Sedlak, A.J. (1974) "An Investigation of the Development of the Child's Understanding and Evaluation of the Actions of Others", Ph.D. Thesis, Rutgers University
- [35] Sridharan, N.S., "The Frame and Focus Problems in AI: Discussion in relation to the BELIEVER System. Proceedings of the Conference on Artificial Intelligence & the Simulation of Human Behavior, Edinburgh, July 1976.
- [36] Srinivasan, C.V. (1973) "The Architecture of a Coherent Information System: A General Problem Solving System", Proc. of the 3rd IJCAI, August 1973.

- [37] Tucker, S.S. (1974) "Cobalt Kinetics in Aquatic Microcosms", Ph.D. Thesis, Rutgers University.
- [38] Vichnevetsky, R. (1973) "Physical Criteria in the Evaluation of Computer Methods for Partial Differential Equations", Proc. 7th International AICA Congress, Prague, Sept. 1973; reprinted in Proc. of AICA, Vol. XVI, No. 1, Jan. 1974, European Academic Press, Brussels, Belgium.
- [39] Vichnevetsky, R., Tu, K.W., Steen, J.A. (1974), "Quantitative Error Analysis of Numerical Methods for Partial Differential Equations", Proc. Eighth Annual Princeton Conference on Information Science and Systems, Princeton University, March 1974.
- [40] Weiss, S. (1974) "A System for Model-Based Computer-Aided Diagnosis and Therapy", Parts I and II, Ph.D. Thesis, Rutgers University; also Computers in Biomedicine TR-27, Feb. 1974.

IV.B INFORMAL PROJECTS

The following is a summary of the various "pilot" projects which have been admitted to SUMEX on a temporary basis pending development of a formal proposal. Many of these projects reflect initial efforts at formalizing analyses of experimental situations in preparation for the development of DENDRAL-like heuristic inference generation and modeling.

IV.B.1 STANFORD PILOT PROJECTS

IV.B.1.a AI IN MOLECULAR GENETICS - MOLGEN

THE MOLGEN SYSTEM FOR EXPERIMENTAL MOLECULAR GENETICS

Prof. J. Lederberg
Stanford Department of Genetics

The MOLGEN system is designed to aid the experimental molecular geneticist in many important phases of laboratory investigation. It will be composed of three major interacting parts: an experiment planning system, an enzymatic action simulation program, and a collection of knowledge bases containing the rules and heuristics of molecular genetics.

The experiment planning program will collect information about a problem from the user, select an appropriate methodology for solution (information retrieval, simulation, hierarchical planning, or some combination thereof) and then work interactively with him to solve the problem. Some examples of the range of problems MOLGEN's experiment planner will deal with are:

1. The user wishes to know which enzymes will function under a given pH or salt concentration--a straight information retrieval problem.
2. The user wants an accurate prediction of the ratio of linear to circular DNA after application of ligase to a given starting concentration of "sticky-ended" DNA--a problem best handled by a discrete simulation.
3. The user wants a verification that a proposed experiment will produce something like a desired result--probably a combination of retrieval and simulation.
4. The user wants a plan to synthesize and then analyze a new DNA structure--a deep problem involving hierarchical planning methods making full use of all program facilities.

The simulation program will provide detailed modeling of the action of enzymes on nucleic acid structures. The program has been shown to

produce accurate and reproducible results on several diverse structures for simple ligation, and is being extended to other common enzymatic actions (exo and endo-nucleases, polymerase, etc.).

The knowledge bases will be composed of collections of the rules and heuristics used by geneticists, as well as facts about enzymes, experimental methods, and physical processes like de/renaturation. They will be designed to allow access in retrieval, simulation, and planning modes, so the major problem lies in representation of diverse types of knowledge in a common, consistent fashion.

Along with the major system components discussed above, certain themes remain dominant in all phases of system design. Primary consideration is given to making the system an easy and natural tool for the molecular geneticist to use. Nucleic acid structure entry, editing, and display is by way of an interactive, user-oriented program. Explanation facilities (in the manner of the MYCIN system) will be provided whenever possible, and all knowledge bases made easily extendable and modifiable. We consider the trust and cooperation of the expert user vital to continued system development, and consider the best way to provide for this cooperation is to make the system immediately useful and intelligible to geneticists.

IV.B.1.b BAYLOR-METHODIST CEREBROVASCULAR PROJECT

BAYLOR-METHODIST CENTER FOR CEREBROVASCULAR RESEARCH
DATA SERVICES RESEARCH LABORATORY

John L. Gedye, M.D.
Department of Neurology
Baylor College of Medicine
Houston, Texas

During the year the data services research laboratory has had a total of about 3,000 hours of man-effort available, of which about 50% has been devoted to implementation of the local facilities described below, and a further 5% has been devoted to the SUMEX pilot study.

A) GENERAL GOALS

Clinical research in neurology, as exemplified by the program of the Baylor-Methodist Center for Cerebrovascular Research, creates a large number of data handling problems of a wide variety of types. The Data Services Research Laboratory seeks to support the program of the center by developing and making available a comprehensive range of data acquisition storage, processing and display techniques for the center's investigative laboratories. These techniques are being designed to facilitate the systematic study of inter-relationships between the different types of data gathered from the various cerebrovascular disease patient groups being studied by the center.

Technical Resources

At the beginning of last December the Data Services Research Laboratory accepted delivery of a Digital Equipment Corporation (DEC) PDP-11/35 computer configuration with 32K core, 2 RK05 disks, a TS03 magnetic tape unit, a Terminet 1200 printer acting as console and hard-copy device, and 2 modified Hazeltine 1200 video display terminals for general interactive use. The system has incoming (1) and outgoing (1) 300 baud modem interfaces to the public switched network the latter incorporating a Bell 801C autodialler. This configuration supports time-shared services, both interactive and batch, based on a single user-language (this is an extended basic called BASYS - the system is currently operating under the commercially supported version of BASYS V3P, known as AIMS V3P - for details see the latest edition of the AIMS-11 programming manual, November 1975, obtainable from ARBAT Systems Limited, 61 Broadway, New York, N.Y. 10006).

Access to SUMEX

This has been by means of TYMNET, which we access through one of the Houston TYMSAT's. At the beginning of the project we used a 300 baud TI

Silent 700 terminal in the normal manner, but since the installation of our local PDP-11/35 configuration in December, we have taken advantage of our autodial facilities and the supporting facilities provided in BASYS V3P and have used our BAYSYS terminals for this purpose. As a result of this experience we are now considering implementing software which will allow an easier interface between our local system and the resources of SUMEX. We have in mind an ability to create files on our system and pass them to SUMEX and to pick up files from SUMEX and store them locally. It is felt that implementing such facilities will greatly facilitate interaction with the SUMEX resource and will lead naturally to the procedures needed to support our AI research.

B) MEDICAL RELEVANCE

The system designed and implemented for the Center for Cerebrovascular Research (BAYSYS - not to be confused with BASYS) allows:

- 1) Maintenance of an immediately accessible, up-to-date, cross-referenced directory of all patients who have, at any time, come under the care or investigation of members of the clinical staff of the Department of Neurology, together with a record for each showing what data has been gathered and where it is located. On March 31, 1976 the directory contained 570 entries, and experience to date suggests that in order to keep up with the patient throughput of the center, new names will be added at a rate of about 100/month. As presently configured the system has a directory capacity of 6,000.
- 2) Storage in a readily accessible, computer-compatible form, of all data gathered on patients which may be relevant to the current and future research interests of the center. Investigatory data is regularly archived on industry compatible magnetic tape in a format which allows subsequent collation using standard sort and merge software.

The work of the Data Services Research Laboratory is organized around the assumption that the research activities of the center can, for all practical purpose, be regarded as a set of inter-related projects, each of which includes planned data acquisition by one or more of the investigative laboratories of the center in accordance with a predetermined schedule.

As a result of providing these primary data gathering services to the investigatory laboratories of the center, the Data Services Research Laboratory will acquire access to a reliable data base covering in principle, the entire range of activities of the center, and this will allow a range of secondary data handling activities to be undertaken on behalf of the center.

It appears that the main technical problem that will have to be solved before it is possible to keep up with the potential flow of data is the design and implementation of a suitable range of data input procedures to cope with the wide variety of data types. It is hoped that the new hand held OCR wand recently developed by Recognition Equipment, Inc. of

Dallas, Texas will allow us to develop a suitably flexible data entry work station for our purposes.

C) PILOT STUDY

The aim of this study has been to formulate a project relevant to the activities of the center which will provide an acceptable and legitimate "point of entry" for artificial intelligence research, and which will allow the systematic formulation of objectives for the future. We are, at the present time, focussing on situations in which a researcher working in the Center for Cerebrovascular Research is required to respond to information from a new source and in some way incorporate it into his understanding of a class of clinical situations.

Background

A continual source of background guidance for our work has been the writings of Stephen Toulmin, particularly his book "Human Understanding" (the first volume of which appeared in 1972) in which he develops a evolutionary approach to the subject in terms of a "populational" account of conceptual change in intellectual disciplines. From the standpoint of Toulmin's approach "men demonstrate their rationality not by ordering their concepts and beliefs in tidy formal structures, but by their preparedness to respond to novel situations with open minds - acknowledging the shortcomings of their former procedures and moving beyond them". The emphasis in his approach to rationality is thus on "change", on the circumstances under which and the means by which men change their concepts and beliefs. Our work to date can be thought of as an attempt to model this approach to the growth of human knowledge in a specific situation - assimilating the results of the new 133XE inhalation regional cerebral blood flow assessment technique. The approach requires at least 3 levels of activity:

1. Choice of goals of rational enterprise
2. Development of concepts
3. Formulation of arguments

The basic approach has been to design a system which will, when provided with a set of descriptions of paradigmatic patients representing two clinical conditions, automatically formulate an optimal algorithm (or in other words a set of decision rules which makes the best use of the available information) for discriminating between those two conditions, and which can then be used on a new set of patients for various purposes. The approach has been tested on regional cerebral blood flow data by checking an algorithm developed from a representative set of 32 patients who acted as paradigms on a similar set of 32 patients and a diagnostic success rate of 90% was obtained in relation to the request "Tell me, on the basis of regional cerebral blood flow measurements alone, whether this individual is normal or abnormal".

The approach appears to have a wide range of applications in the context of the work of the center and effort is currently being concentrated on applying the technique to the systematic exploration of regional cerebral blood flow differences in relation to such contrasts as "male/female", "left-hemisphere/right hemisphere" and so on. The technique can be applied to data as it accumulates, thus allowing the detection of trends at an early stage of the research.

It is inappropriate to go into details of the approach here, but it's essential feature is that it permits revision of both conceptual boundaries (such as what is meant by "high" as opposed to "low" flow in a particular brain region) and of arguments expressed in terms of a given set of concepts (such as: "high" flow in region x, together with "low" flow in region y, and "high" flow in region z implies multi-infarct dementia as opposed to Alzheimer's dementia) as a result of the acquisition of new data.

II) INTERACTIONS WITH THE SUMEX-AIM RESOURCE

A) Collaborations Through the Network

We have not yet reached a stage at which we are able to support regular collaboration through the network. We hope that this will develop naturally as soon as we have been able to develop interface between our local PDP-11/35 system and SUMEX so that we can handle interaction as a natural part of our day to day activities. Dr. David Bowen is planning to cooperate with us during the summer from London over ARPANET, and we intend to work together on his neurochemical data.

B) Contacts and Cross Fertilisations

The Rutgers workshop was the most valuable feature of SUMEX-AIM participation during the year, providing an opportunity to get an overview of work in progress and to see directions in which our work here could develop to complement what was being done elsewhere. On the clinical application side the "INTERNIST (DIALOG)" project was the most stimulating, as it demonstrated the challenges that would have to be met by any practically useful approach, for example: the ability to handle multiple problems presenting together.

At a more fundamental level "DENDRAL" confirmed the value of the approach to which we are committed - trying to model the research process itself. As a result of the Rutgers workshop, a working relationship has been established with Drs. Lindberg and Blackwell of the University of Missouri and there have been reciprocal visits between our respective locations. On my last visit to Columbia, I gave an invited paper called "A Jurisprudential Approach to Artificial Intelligence" and have since been invited to write this up for "Biosciences Communications".

Dr. Lindberg's work on the encipherment of electrolyte patterns has proved to be a useful stimulus to our own work on the encipherment of

neurological data, and suggestions from myself that it might be valuable to look at electrolyte pattern transitions has been taken up and developed as an application of the theory of finite state machines.

C) Critique of Resource Service

Our use of the computational resources of SUMEX has so far been largely confined to experimentation with the various modules of the system described above. This was particularly valuable before our own resources became available. We now see ourselves moving to a new mode of operation in which we try to find out which things are best done on SUMEX and which locally. Our main criticism to date has been slowness of response during peak hours, when, unfortunately we have sometimes had to try to operate because of constraints on manpower availability.

FUNDING STATUS

Work is currently being supported by departmental funds. However, we have recently received unofficial notification from NIH that funds have been approved for the support of the Data Services Research Laboratory in the center grant renewal effective February 1, 1977. Approval is for one year in the first instance with support for a further two years subject to satisfactory administrative arrangements.

IV.B.1.c AUTOMATIC LV MODELING

Automatic Radiographic Image Analysis by LV Modeling

Donald C. Harrison, Professor of Medicine
Edwin L. Alderman, Assistant Professor of Medicine
Lynn Quam, Ph.D., Research Associate in Computer Science

Stanford University

A proposal to carry out this research has been submitted to the NIH. Medical applications of computer image processing was part of the original collaborative research goals of SUMEX. This has been supported as a pilot project to facilitate the development of independent grant support.

The proposal is to use the facilities of SUMEX-Aim to develop a mini-computer system for the automatic analysis of left ventricular angiography in a clinical cardiac catheterization laboratory setting. This system will be designed to 1) provide frame by frame quantitative volume measurements 2) analyze wall motion abnormalities and 3) generate new information about left ventricular function. In conjunction with the SUMEX systems staff, Dr. Lynn Quam has done the initial development work on an interactive graphics and image display system as summarized below.

SUMMARY:

A general purpose hardware and software system for interactive graphics and grey-level image display has been developed on the SUMEX Tenex system, using a Tektronix 611 storage scope controlled by a PDP-11/10 processor. The system is capable of producing limited displays dynamically using the non-storage mode of the 611, whereas complex displays require the use of storage mode or photography.

A general purpose graphics package has been developed, which is essentially compatible with the graphics software at the Stanford A-I Lab. Consequently, with minor revisions, many of the graphics programs written at the A-I lab can be used at SUMEX.

Many of the image processing algorithms originally developed at the A-I lab by L. H. Quam have been revised to run in the Tenex environment.

The combined effect of the graphics and image display hardware and software is the capability for SUMEX users to perform a wide variety of image enhancement operations on grey-level images, and to display both line drawing graphics and grey-level images.

PURPOSE:

The primary purpose for developing the graphics and image display

system was to support the needs of an NHLI proposal in the division of Cardiology. The proposed research was to develop algorithms to automate the procedure for outlining ventricular margins in angiograms, for the purpose of cardiac dynamic performance evaluation. The images are obtained by passing x-rays thru the patient who has a catheter placed in the heart. The x-ray target is viewed by both a cine film camera and a vidicon. The vidicon output is both viewed directly, and recorded on a video disc. Several cardiac cycles are recorded on the disc, then a radio-opaque dye is injected into the heart using the catheter, and several (at least 3) more cycles are recorded. This procedure produces about 150 images which must be analyzed.

In the normal clinical operation, a technician manually traces the ventricular margins using a light pen which is connected to a mini-computer which computes the desired performance measurements and produces hard copy output. The manual tracing is quite tedious and slow.

The primary difficulty with automating ventricular margin outlining is that during part of the cardiac cycle the margin is of very low contrast (poor signal to noise ratio), making it impossible to detect without taking adjacent (in time) images into consideration. In order to develop techniques for automatic margin definition, it was necessary to have hardware and software for image and graphics display.

HARDWARE:

The hardware consists of four component parts: a Tektronix 611 scope, a display controller, a PDP-11/10, and a PDP-10 to PDP-11 communication interface.

Tektronix 611 Scope:

A Tektronix 611 storage oscilloscope is used to generate the images. Briefly, the 611 scope has high resolution (about 100 points to the inch on a 7 by 9 inch screen), and storage and non-storage modes. Using non-storage mode increases the resolution by about a factor of 2, and allows the the display of grey-level images. Unfortunately, the 611's deflection system is too slow for direct viewing of very large images without an intolerable flicker. For large grey-level images one of two approaches must be used: photographic recording of the 611 screen, or halftone grey-level simulation using storage mode.

Display Controller:

The X, Y and Z axis signals to the 611 scope are generated by a display controller designed at SUMEX. Basically, the X and Y deflection signals are generated by two 12-bit digital to analog converters which are driven by X and Y position registers in the display controller. The Z-axis signal is controlled by a digital level which turns the 611 beam on for a time proportional to the binary number in the Z axis register.

The display controller is capable of two primary modes of operation: vector and raster. Vectors are generated as a sequence of discrete points. Vectors are specified to the hardware by the starting X, Y location, the DX, DY distance between the discrete points of the vector, and N the number of points in the vector. Grey-level rasters are generated as a sequence of discrete points (pixels) each of which can have an arbitrary intensity. To the controller, a single raster line is specified exactly the same as a vector with the addition of N 8-bit bytes of Z-axis intensity information.

The display controller is connected to a PDP-11 Unibus.

PDP-11/10:

A PDP-11/10 minicomputer is used to control the display. For images in non-storage mode, the PDP-11 dynamically refreshes the screen at about 3 microseconds per point (depending on the brightness: 3 microseconds is the minimum time). For halftone grey-level simulation, the PDP-11 executes the halftone algorithm.

PDP-10 to PDP-11 Interface:

A general purpose communication interface connects the PDP-10 to the PDP-11. This hardware consists of two 32-bit registers, one for each direction of data transfer, 2 status registers, and 2 control registers. Using this interface, data can be transferred between the PDP-10 and the PDP-11 at about 20 microseconds per word data rates (potentially).

SOFTWARE:

The software to utilize the display hardware consists of many modules some of which execute on the PDP-10 and others on the PDP-11.

Communication Module:

Communication between the PDP-10 and the PDP-11 is accomplished by transferring blocks of data thru the communication interface which is controlled by programs running in each machine. The basic operations are:

- a. Load a program in the PDP-11
- b. Send a block of data to the PDP-11
- c. Get a block of data from the PDP-11
- d. Start a program running in the PDP-11
- e. Stop the program running in the PDP-11
- ... and a few other operations

PDP-11 Display Module:

The display module interprets blocks of data sent thru the communication interface as commands to the display. The basic display commands are:

- a. move the beam to position X,Y
- b. draw a line consisting of N points, incrementing the beam position by DX,DY between each point.
- c. generate a grey-level raster
- d. generate a half tone raster
- e. set beam brightness
- f. display a string of text
- g. display subroutine call
- h. display subroutine return

From these primitive commands all higher level display functions are built.

PDP-10 Display Module:

The PDP-10 display module interprets SAIL procedure calls and produces blocks of data to send to the PDP-11 display module. In addition to the primitives listed above, many higher level display functions are implemented:

- a. display a circle
- b. display an arc of a circle
- c. plot a graph of the data in a array: labelling the axes

PDP-10 Image Processing Functions:

Many of the image processing functions developed at the Stanford Artificial Intelligence Laboratory have been modified to run under Tenex. The following is a partial list:

- a. Input an image from a disk file
- b. Output an image to a disk file
- c. Input a window of an image from a disk file
- d. Display an image on the 611 scope
- e. Enhance the contrast (stretch) of the image
- f. Rotate the image 90 degrees clockwise
- g. High-pass filter the image
- h. Low-pass filter the image
- i. Display the histogram of the image
- j. Expand the image
- k. Remove "noise" from the image (local sigma test)
- l. Difference two images

IV.B.1.d INFORMATION PROCESSING PSYCHOLOGY PROJECT

INFORMATION PROCESSING PSYCHOLOGY

Prof. E. Feigenbaum (Computer Science)
and Prof. H. Cohen (U. C. San Diego)

May 1976 Report

Information Processing Psychology is concerned with the construction of models of human cognition, using the methodologies of computer simulation and artificial intelligence. The attempt is to give a precise characterization of the human information processes and information structures that underly human problem solving, learning, and perceptual behavior. Over the past two decades, research in this scientific area has produced computer models of behavior in puzzle-solving, game-playing, and theorem-proving tasks; rote learning laboratory tasks; linguistic understanding and long-term memory tasks; pattern extrapolation tasks, e.g., as are found in intelligence tests; children's seriation tasks; concept attainment tasks; visual scene understanding tasks; tasks involving mental imagery; and many others. A type of human cognitive/perceptual activity that has not been much studied is the behavior associated with the production of works of art. In the past, neither graphical/visual art-making nor musical composition has been studied in depth.

The particular project described below has sought to bring under examination one of the least well-defined areas of higher intellectual functioning -- the activities of art-making performance -- and to develop a computer model (i.e., information processing model) capable of verifying the plausibility of a number of hypotheses concerning such activities. We have addressed the subset of art-making behavior which is concerned with the production of freehand drawings, and in particular drawings which might be characterized by their imagistic richness as opposed to formal complexity.

The computer model has followed the format used in many A.I. programs: a production system in which an explicit body of knowledge is encoded as a set of rules linking the recognition of complex prior program behavior (in the making of the drawing) and current states within the drawing itself, to the exercise of appropriate subsequent rules, which in turn move the drawing into new states. The model is to be regarded as an expert or specialist, in the sense that the encoded knowledge is specifically concerned with the mechanics of image-building and does not encompass any other aspect of the world.

Since much of what we understand by "meaning" in images -- as elsewhere -- clearly involves world knowledge, there may seem to be something anomalous in a program without world knowledge designed to generate imagistically rich drawings. However, our belief has been that a large part of "meaning" is signalled by the image-structure itself, and that this is related more to the nature of underlying perceptual processes than to any particular stored perception of the world. There should be a

set of pre-acculturated behavioral patterns of so fundamental a kind that their very exercise would persuade the viewer that some "meaning" was intended.

Following from this position, our selection of appropriate production rules in the production system has tended to stress a number of low level perceptual activities. Early versions of the model were able to differentiate figure from ground, closed forms from open forms, inside from outside; and also to perform tasks -- like generating a path from one point to another under certain constraints -- in feedback mode, which required a continually updated model of the state of the drawing under construction.

More recently, the model has been given enough knowledge of the mechanics of representation to permit it to manipulate the emerging drawing more fully. Thus, it knows that a closed form may function as a delineated area upon which other markings may be made; or whose flatness may be stressed by cross-hatching; or that the form may "stand for" a solid object which may be shaded or cast shadows.

The protocols referred to above are families of behavioral rules which are distributed throughout the system and become enmeshed into complex structures. For example, one aspect of the model's awareness of figure-ground relationships is a set of avoidance protocols, which prevent the invasion of existing elements in the drawing. Which of the set will be invoked will depend upon both what is being done -- in terms of currently "open" protocols -- and what is being avoided.

The major protocols currently available to the model may be summarized as follows:

closure: forms may be closed by preplanning or, at a later stage, as suggested by the state of the drawing. Reinforced by hatching, shading, marking (recursive repetition, see below), piercing, accretion.

placement: the model is able to select unused areas of the drawing of a shape and size appropriate to its current plan; its subsequent behavior is then determined in large part by the precise consideration of its environs. The model has no aesthetic criteria or compositional strategies beyond providing itself with adequate space.

avoidance: may result in the discontinuation or the modification of the current plan, with or without the development of an alternative plan; or in an attempt to circumvent the obstructing form.

repetition: in the "placement" area of the production system, this would result in similar sub-histories being repeated, subject to local conditions, in other parts of the drawing. In other conditions it will result in a recursive use of closed forms as fields upon which other closed forms may be made; in multiple division or extension of an area; in zigzags or groups of parallel lines, or in concentricity.

Long term plans include the provision of simple "world knowledge" to the problem, in order to investigate plausible specialist/non-specialist

interactions in the drawing process as a source of imagistic richness. We have done some recent experimentation with people, designed to isolate and examine the protocols actually employed by a group of drawing students in a visual arts class at U.C., San Diego. The results of these experiments are now undergoing statistical analysis, and it is anticipated that much useful material will be available for the next stage of program development.

The program described above was developed in SAIL on the SUMEX facility, partly during the period when Professor Harold Cohen, of the Visual Arts Department of U.C., San Diego, was on leave at Stanford University, and partly upon his return to his campus. He has assumed the Directorship of a new Project for Art/Science Studies at UCSD, and has by gifts and grants procured for the Center a PDP/11-type facility capable of supporting the research described above on the modeling of art-making behavior. The innovative work of this SUMEX pilot project will therefore be "spun off" to a fruitful environment at UCSD. As a SUMEX activity, this particular research has effectively terminated.

Bibliographic References

1. Cohen, H. "Steps toward a Theory of Meaning", invited paper, International Sculpture Conference, U. of Kansas, March, 1974
2. Cohen, H. "On Tools and People, Including Computers and Artists", invited paper, Conference on Computers in Art, Purdue Univ., 1975.
3. Cohen, H. "The Simulation of Perception: Problems in Generating Drawings by Machine", invited paper AAAS Annual Meeting, 1972.
4. Cohen, H. "On Purpose", <<Studio International>>, January, 1974.
5. Cohen, H. "Parallel to Perception", <<Proceedings of the Edinburgh Conference on Art and Computing>>, 1973

IV.B.1.e AIM RESEARCH - UNIVERSITY OF ROCHESTER

AIM Research - University of Rochester

Drs. Feldman, Rovner, and Low
Rochester University

(Grant NSF DCR74-24203, 2 years, \$149,956 total and
Sloan Fdn. 74-12-5, 3 years, \$120,000 last year)

SUMEX facilities are being used at the University of Rochester by a group of about 10 second-year medical students under the direction of Charles Odoroff, Biostatistics. Their work is either an optional part of a course in biostatistics and epidemiology, or an individual project. They have studied some of the documentation of the MYCIN system, and have experimented with using it both on canned case histories and on cases from the files of microbiologists at the U. of R. Medical School. At least one evaluative paper has been written. It is planned to use the CASNET system in the same way.

There is continuing system development work, especially for the SAIL language system.

IV.B.1.f QUANTUM CHEMICAL INVESTIGATIONS

QUANTUM CHEMICAL INVESTIGATIONS OF
HEME PROTEINS AND FERREDOXINS

Dr. Gilda Loew
Stanford Department of Genetics

(Grant NSF GB-40105, 2 years, \$18,000 this year)

SUMEX is used for the calculation of various one-electron electromagnetic properties of iron containing compounds. The programs were formulated and written by David Steinberg, Michael Chadwick, and David Lo. David Lo was responsible for converting the programs for interactive use on the PDP system. Slight improvements were made by Robert Kirchner, and Sheldon Aronowitz is currently expanding the formulation to include additional spin and oxidation states of the iron atom.

The properties that are calculated include the electric field gradient at the iron nucleus, quadrupole splitting, isotropic and anisotropic hyperfine interaction, spin-orbit coupling and zero field splitting, g values, and temperature dependent effective magnetic moments. The calculated values are compared directly to experimental results obtained from published Mossbauer resonance and electron spin resonance spectra. Such a comparison determines not only the reliability with which these properties can be calculated but also gives an indication of the ability of the model of the iron active site to mimic the actual environment found in a particular compound or iron containing protein.

The major input to these properties programs is a description of the electron distribution of the compound under consideration. This description is obtained using a semi-empirical molecular orbital method employing the interactive extended Huckel procedure. Such a calculation requires up to 660K core and is performed elsewhere. When the calculated electron distribution yields a set of calculated properties in agreement with observation, we have increased faith in the description of the model of the active site and can carry the model one step further to make qualitative inferences about certain properties relevant to the biological functioning of the compound. These properties, which are harder to characterize experimentally, include the nature of the ligand binding to iron, relative bond strengths themselves, net atomic charges, and electric potentials. The model may be varied (that is, change the spin or oxidation state of the iron, replace certain ligands, or simply change the geometry of the ligands) and a new set of properties calculated to predict what effect these changes would have on the observed electromagnetic properties.

Such a procedure lends itself well to the study of three classes of iron containing compounds of biological interest: the one-iron sulfur proteins known as rubredoxins and the two-iron sulfur proteins called

plant-type ferredoxins which serve as one electron transfer agents, heme proteins which serve as oxygen as well as one electron transfer agents, and sideramines which serve as iron transport agents. The calculated properties for the first class, used to elucidate the geometry of the sulfur ligands and the spin state of the iron within the protein, are reported in the following

PUBLICATIONS:

- [1] G.H. Loew, M. Chadwick, and D.A. Steinberg, Theoret. Chim. Acta (Berl.) 33, 125 (1974)
- [2] G.H. Loew and D.Y. Lo, Theoret. Chim. Acta (Berl.) 33, 137 (1974)
- [3] G.H. Loew, M. Chadwick, and D.Y. Lo, Theoret. Chim. Acta (Berl.) 33, 147 (1974)
- [4] G.H. Loew and D.Y. Lo, Theoret. Chim. Acta (Berl.) 32, 217 (1974)

We are currently performing a systematic study of heme proteins. The electromagnetic properties of these proteins and of synthesized compounds which mimic the observed behavior of the proteins have been well studied experimentally. But many questions regarding the nature of small ligand binding to the heme group remain unresolved. Before we can address ourselves to such problems, we must first be able to theoretically reproduce the experimentally observed behavior. The specific areas of interest are:

- a.) deoxy heme in both the relaxed (iron in the plane with a low or high spin state) and tense (iron out of the plane with a high spin state) configurations.
- b.) oxy heme with various oxygen geometries (co-axial or coplanar) and several excited electronic states (promotion of an electron from an iron d-orbital to an unfilled oxygen orbital).
- c.) abnormal heme compounds which do not bind oxygen but which bind axially to CN, N₃, NO, or OH.
- d.) the enzymatic cycle of cytochrome P450 camphor, in which the protein has been isolated with the iron in various spin and oxidation states.

Preliminary results for oxy heme have been published (Loew and Kirchner) in the J. Amer. Chem. Soc. 97, 7388 (1975).

We have also calculated the electromagnetic properties of ferrocene Fe(C₅H₅)₂ and the binuclear transition metal complexes biferrocene and biferrocenylene in various oxidation states. The work has been published (Kirchner and Loew) in Theoret. Chim. Acta (Berl.) 41, 1 (1976) and submitted (Kirchner and Loew) to Inorganic Chemistry.

The heme work is funded by the National Science Foundation Grant GB 40105, which was renewed starting June 1976 for a period of two years.

Undergraduate research projects which attempt to correlate reactive electronic sites for a series of polycyclic aromatic hydrocarbons with carcinogenic activity use SUMEX to calculate various measures of C-C and C-H bond reactivity. Again, the major input to this program is taken from the results of an iterative extended Huckel molecular orbital calculation performed elsewhere. The only current funding for these projects is a SCIP computer processing subsidy although an application to the National Institute of Health is pending.

IV.B.2 NATIONAL PILOT PROJECTS

Over that past year several national pilot projects have been initiated with the approval of the AIM Executive Committee and advice of the AIM Advisory Group. One of these (the ACT Project under Dr. John Anderson) has moved from pilot status to become a formal project. The currently active pilot efforts are summarized below.

IV.B.2.a NATURAL LANGUAGE UNDERSTANDING

Natural Language Understanding

Prof. R. Lindsay
University of Michigan

(Financial support from University of Michigan)

I. Summary of Research Program

The major aims of this pilot project have been to establish research goals, initiate collaborations with faculty at the University of Michigan Medical School, and to develop software. The project staff consists of Associate Professor Robert K. Lindsay, Dr. Maija Kibens (Research Associate), and Mrs. Kathie Gourlay (Programmer Analyst), all of the University of Michigan.

A) Technical Goals

The overall goal of this project is the development of a histological model that will assist anatomists in the design and evaluation of methods of organ culture. As conceived at present, our technical goals are the design of (a) a data structure for encoding descriptions of microscope slides made from organ explants, (b) a model of microanatomical processes based on the expertise of histologists and pathologists, and (c) means for construction of the data structures of (a) from histologists' verbal descriptions.

B) Medical Relevance and Collaboration

The value of such a system to biology and medicine is far-reaching to the extent that it succeeds in assisting in the development of organ culture methodology. To illustrate with a single important example, the ability to cultivate the organs of experimental animals is the first step in the in vitro study of disease processes such as cancer in those organs. We are working in collaboration with three anatomists in the Department of Anatomy, Professor Raymond Kahn, Associate Professor William Burkel, and Assistant Professor Theodore Fischer. For the past two years this group has been experimenting with methods for cultivating canine prostate.

C) Progress and Accomplishments

Our efforts have been directed along two fronts. We are familiarizing ourselves with the current capabilities, knowledge, and problems of the histology group. We are also developing artificial intelligence programs to understand histological information typed in a natural language format.

Collaborating with the histologists

1. We are interviewing the principal investigators and their several assistants individually to learn from each of them his conception of tissue functioning.
2. Formulating better analysis methods - Together with the histologists we have designed a new grading scheme for recording the maintenance status of an organ explant. This scheme is more complex than their previous category system in that it includes more factors and finer distinctions. Currently, the group is using standard non-parametric statistics to compare the evaluations of the explants.
3. Designing an AI model - The histologists are enthusiastic about the new evaluation method. They believe it to be a great improvement over their previous one. However, they recognize its inadequacies and are open to any artificial intelligence techniques that will enable them to capture more of their knowledge about each explant in a form that can be used in the design and evaluation of experiments.

Software Development for Natural Language Input

1. Formulating a general design - The proposed structure for the system includes multiple sources of knowledge, each contributing hypotheses about the meaning of the input. The input is to be freely formatted with possible typing, spelling, and syntactic errors. The output will be an internal representation of the meaning of the input text, namely a representation of an organ explant. The knowledge will include components for: typing correction, word decomposition, morph recognition, syntactic analysis, semantic analysis, and histological knowledge. The knowledge components of the system are being designed to be independent of each other insofar as possible.
2. Implementing - The knowledge components for typing correction and word decomposition have been written in INTERLISP. The dictionary format has been designed. Work is currently being done on the morph recognition component.

D) Publications

A manuscript describing the results of the application of the revised evaluation scheme has been written by Professors Kahn, Burkel, Fischer, and Herwig (the project surgeon). The paper is titled "Effect of

vitamin A on canine prostate in organ culture". There have been no publications to date on the AI aspect of this project.

E) Funding Status

The canine prostate project is funded by the National Cancer Institute. A grant application to NIH for similar work with lung is pending. The salaries and research facilities of Professor Lindsay, Dr. Kibens, and Mrs. Gourlay are provided by the University of Michigan Mental Health Research Institute, a division of the Department of Psychiatry in the Medical School.

II. Interactions with the SUMEX-AIM resource

A) Medical Use of Programs through Networks

We have not had occasion for such use.

B) Useful Contacts and Cross Fertilization with Other SUMEX-AIM Projects

Kathie Gourlay is on the LISP users' mailing list. Most of the other users and personnel at SUMEX have been very helpful in giving advice and solving problems. A few examples of such assistance during the past year follow.

There have been some problems with the speed of response in an INTERLISP program. Masinter has been very helpful with suggestions. In one instance, he was able to run the program and interactively pinpoint some slow spots.

We have also had communication via SNDMSG with N. Smith, Hedberg, Feigenbaum, Davis, Lederberg, R. Smith, Colby, Parkison, Winograd, and others.

Dr. Kibens has used the LINK facility to obtain answers to questions about details of system use, and has used SNDMSG extensively for such purposes. Other interactions have concerned obtaining information about current status of natural language input systems for interactive programs. SUMEX has also been a very convenient communications facility via SNDMSG to non-SUMEX users at SRI, CMU, and SU-AI.

C) Critique of Resource Services

The SUMEX staff has been very helpful. To cite just one example: At one time last year, we wanted to transfer some data to SUMEX from a PDP-9 minicomputer which is located here. Gourlay communicated with Cower via SNDMSG and arranged to mail a DECTape. Contrary to what the DECsystem 10 Assembly Language Handbook led Cower to believe, the PDP-9 DECTape was not compatible with the PDP-10 software. Cower and another person spent

several hours working unsuccessfully to transfer the data. We appreciate their efforts although the problem could not be solved because of serious incompatibilities.

Now that Tymnet has several local lines to our area (since January 1976) and we have terminals located in our offices the SUMEX facility is very convenient. The system is quite reliable. However, when it is down the explanation given us from Tymnet is almost always out-of-date, e. g., maintenance work that was completed hours ago.

There should be a manual such as the Tenex Executive Manual that explains the features available on the SUMEX system. At present, it is necessary to look in dozens of different documentation files or to learn by hearsay. In the interim, it would be good to have one of the system personnel designated as a general source person for details of system use.

We would suggest that some thought be given to making the LINKing facility a more productive and convenient means of communication. While LINKing is a potentially useful device, it is also a potential nuisance to the recipients. This is a cause of our reluctance to use this facility. Perhaps an explicit policy should be decided upon by all SUMEX users to establish what is the community attitude toward LINKing. It might facilitate communication by LINKing if the default option were changed to REFUSE, but modified to allow immediate acceptance of the LINK upon learning who it is from. Certain programs, such as TYPE, are usually run in REFUSE mode. Perhaps these programs should set REFUSE mode on entry and clear it on exit so that the user would be protected from interruption at those times without needing to have the REFUSE mode set permanently. There are obviously many such technical changes that could be made to improve this feature, and we would like to see some discussion of them.

We look forward to the availability of 1200 baud capability over TYMNET so that listings can be obtained more rapidly. Any encouragement from SUMEX to TYMNET that would speed the conversion would be appreciated. We think it would be wise for the system to be compatible with the VADIC protocol (soon to be adopted by Bell, we hear unofficially) rather than with the troublesome Bell 202 equipment, as announced.

IV.B.2.b KRL PROJECT

Knowledge Representation Language - KRL

Dr. Dan Bobrow, Xerox PARC
Dr. Terry Winograd, SU AI Lab

This pilot project was just initiated on the SUMEX-AIM facility and is a medicine-oriented extension of the KRL development effort at Xerox Palo Alto Research Center. The basis of the original project is the development of a systematic programming framework within which to describe and manipulate knowledge about a task domain and which may be used by a performance program to reason and solve problems within that domain. The development of such AI tools is an important part of the AIM community in that it allows the more coherent and general formulation of medical AI programs.

A first version of KRL has been implemented and several students will experiment with implementing medical consultation programs (e.g., MYCIN, CASNET, or Rubin's model of renal disease) using KRL.

IV.B.2.c COMPUTERIZED PATIENT MONITORING

Computerized Patient Monitoring and Clinical Decision Making

John J. Osborne, M.D. Director Intensive Care Services
Richard R. Mitchell, Ph.D. Biomedical Engineer

The Institutes of Medical Sciences
Pacific Medical Center

The immediate desire of this pilot project is to use MLAB and to explore the opportunity to become a regular member of the SUMEX-AIM user community.

The project is part of a Bioengineering and Computer Science Resource for medical research. The Research Data Facility of the Institutes of Medical Science is a NIH funded center for the development of computerized patient monitoring and clinical decision making. The emphasis to date has been in the area of clinical monitoring of the respiratory parameters of critically ill patients.

There are three major areas of potential joint cooperation with SUMEX-AIM:

1. Clinical Decision Making;
2. Image Processing;
3. Modeling.

In the area of Clinical Decision Making the project is funded to develop an intelligent system for detecting and reporting alarm conditions in the hospital intensive care environment. Its goals include using sophisticated image processing techniques for the evaluation of pulmonary physiology using the scintillation camera and ¹³³Xenon. The present work in modeling is limited by the capabilities of the IBM 1800 CSMP and the investigators are interested in exploring the use of more complicated models requiring a sophisticated simulation language.

IV.B.2.d AI IN PSYCHOPHARMACOLOGY

Artificial Intelligence in Psychopharmacology
(NIH grant application in preparation)

Dr. Jon F. Heiser, M.D.
Assistant Adjunct Professor
Dept. of Psychiatry and Human Behavior
University of California at Irvine

A. Introduction

This project has just been authorized as an AIM project. The following quote from a letter of Drs. Buchanan and Axline of the MYCIN project describes the collaboration that led to their project and which is expected to continue.

"He is extending MYCIN's knowledge base to cover consultations regarding chemotherapy for psychiatric disorders. This is valuable to us for at least two reasons: it increases the potential uses of the program and it illuminates those specific parts of the program that are not yet general enough to be easily extended to new areas. By pioneering in the effort to develop a more general framework for medical reasoning computer programs, Dr. Heiser is helping us provide a means for encoding and testing large amounts of medical knowledge."

The objective of the new project is to develop computer based automated systems capable of assisting in research, teaching and consultation in psychopharmacology. It will result in the development of software which will run on the University of California, Irvine PDP-10.

[The following material is abstracted from Dr. Heiser's proposal to the AIM Executive Committee].

A.1 BACKGROUND:

Information in medicine expands so rapidly that both researchers and clinicians struggle to digest it and apply it wisely. Computer-based instruction (textbooks, journals, individualized supervision and consultation) is one solution to this problem. By their very nature computer-based systems can be programmed to 1) explain their reasoning in natural language and in terms intuitively acceptable to users of various degrees of sophistication, 2) have their behavior totally analyzed, and 3) be easily modified or updated.

Computer-based knowledge systems have been developed for describing and solving problems in pharmacology. At Stanford University Medical Center in California, when new drugs are prescribed for a patient, their profile of action is compared to that of drugs already consumed by the patient. A warning is generated for the physician if a potential interaction is noted (1). Artificially intelligent systems have been

developed which utilize pharmaco-kinetic models to suggest initial doses and monitor on-going maintenance doses with complex drugs such as digitalis (2). Artificial intelligence systems are also being generated to diagnose patients with infectious diseases and to suggest appropriate antibiotic agents (3).

Several other systems were discussed at the First Annual AIM (Artificial Intelligence in Medicine) Workshop, held at Rutgers University 14 June through 17 June 1975. (S. Amarel and C.A. Kulikowski of the Computer Science Department at Rutgers University directed the conference).

We have begun to adapt the techniques of the Stanford Group (3,4) in the generation of an artificially intelligent system which evaluates and diagnoses psychiatric patients, suggest pharmacological treatment and monitors the on-going clinical course.

The system has 16 rules, based on conventional clinical observations, for diagnosing either mania or schizophrenia. Clinical findings are collected and a diagnosis made by manipulating human expert generated "certainty factors" which are similar to but not identical to probabilities. their precise mathematical nature and manipulation are described in Shortliffe and Buchanan (4).

A.2 RATIONALE:

Psychopharmacological agents are frequently misused qualitatively and quantitatively by prescribing physicians as well as by consumers. Consultation with experts in psychopharmacology is frequently sought and is given on the basis of clinical data, currently established practice, evolving research or ad hoc hypotheses. A computer based consultation system, available 24 hours per day, could greatly assist non-specialist physicians in choosing the best psychopharmacological treatment, given the same expertise and data. Such a system could also serve as a teacher-advisor to students and as a reference for various types of psychopharmacological knowledge, e.g., well established principles and practice, new but not fully verified ideas and late breaking developments (5). Properly weighted, all such information could be used in consulting and teaching functions. For example, the system could suggest less well established, more controversial or more hazardous diagnostic or therapeutic techniques for a patient with a life-threatening situation not responsive to conventional measures. Like the human clinician, in desperate or excessively chronic circumstances, the system could generate novel hypothesis with an estimate of potential risks and benefits.

B. SPECIFIC AIMS:

1. To study existing automated systems, computer based and otherwise, which assist in clinical decision making.
2. To develop a model of expert clinical decision making for clinical psychopharmacology.

3. To implement this model on a computer system such that the system can converse in real time in natural language through computer terminals with users located close by or remotely.
4. To evaluate the performance of the system as a teaching and consulting aide.
5. To increase the breadth and depth of the artificially intelligent system by a) increasing the technical sophistication, e.g., by adding options such as voice activated microphone-loudspeaker (or print-out) terminals. b) adding other areas of clinical psychiatry towards an ultimate goal of having a fully automated and self-contained textbook-consultant for psychiatry. c) linking the system to precoded data bases so that the system could quickly learn from thousands of actual case histories and use this data and experience both to modify its intuitively human-like rule-based decision model and to generate abstract mathematical or statistical decision making models. d) integrating the system with biomedical data collecting techniques achieve more direct involvement in research and patient evaluation. e) proposing new drugs to be synthesized or tested for desired psychopharmacologic affects.

Aims 1-4 should easily be attained within five years. Aim number 5 is obviously quite speculative and dependent on developments in computer technology, biomedical engineering, etc. Promising beginnings have been made. An example is a program which reads and analyzes the content of typed scripts of spontaneous human speech (6). This system parses sentences into noun and verb clauses, recursively or repeatedly if necessary, and uses a set of rules to score the noun and verb phrases for a variety of affects and states of mind by means of a well documented, reliable and valid technique of content analysis otherwise requiring a human content analyzer with a common sense knowledge of the world and the language.

C. METHODS AND PROCEDURES:

We plan to study existing systems and to develop a similar system in clinical psychopharmacology. Dialogues, equivalent to those planned for this development, are routinely produced by the antibacterial program mentioned above in (3) and could be available in the area of clinical psychopharmacology within a year in complete enough form to be evaluated. This work will be done in consultation with the Information and Computer Science Department at University of Calif., Irvine and the Computer Science Department at Stanford University. During approximately the first two years efforts will be concentrated on developing the artificial intelligence techniques referred to above, including question answering in natural language, abstract reasoning, advice giving, etc. The expert information will be installed in sentence-like form initially from the working knowledge of the principal investigator and the behavior of the system compared to the behavior of the principal investigator. In the second phase expert information will be abstracted from standard textbooks, journals and consultation with acknowledged experts. Here evaluation becomes more difficult except when the system makes an obvious

mistake. A third level of expert input will include other and possibly non-human information sources such as actuarial formulas and other statistical techniques (12).

In later phases of the project more concern will be placed on the diagnosis problem. This problem is being deemphasized during initial phases because it has received reasonable attention by other groups (7-12).

Many of the above mentioned systems have been formally or informally evaluated and found to perform, within their range of applicability and our ability to measure performance, as well as acknowledged experts (12). Consultation with experts in evaluation research in both education and clinical medicine is available locally and will be enlisted in later aspects of the project once a workable system has been developed.

Risks and hazards in this procedure are minimal since no biological material is involved, no patient records are used and identification of patients to be discussed can be secured or eliminated with no impact on the system or the user. Use of a consulting and teaching system in clinical psychopharmacology might involve clinical responsibility and could be regarded as contributing to or responsible for an error of omission or commission in clinical judgement and practice. Every attempt will be made to complete a thorough evaluation of the system, its validity and reliability before it is made available to other than a small testing group. If in later phases we add the capability of utilizing large data bases such as available through the Missouri Information System (10,12) only those well developed procedures for transmitting large data bases with complete anonymity and protection of individual patient rights will be used. Such procedures will be rigidly and consistently adhered to in all aspects of this project.

D. SIGNIFICANCE:

It is hoped that results from these initial studies will stimulate further research by physicians and graduate students in related fields such as biochemistry, pharmacology, pharmacy, mathematics and the information processing sciences. To have instant access to teaching and consultation based on a consensus of the best data, the best abstract mathematical or statistical "number crunching" techniques and the best human experts would be of great value to researchers, specialists, family physicians, students and others. Evaluation of the effects of such a system and comparison with traditional methods of research, teaching and consultation would be of great benefit to medical educators. It is hoped that many students would master, beat or "psyche out" the system. This would be excellent evidence that learning is occurring. However, because of the information explosion, periodic updates to and from the system should prevent it from becoming obsolete.

References

1. Cohen, S.N. et al. Computer-based monitoring and reporting of drug

- interactions. Proceedings MEDINFO IFIP Conference, Stockholm, Sweden, August 1974.
2. Silverman, H. A digitalis therapy advisor. MAC TR-143, Massachusetts Institute of Technology, Cambridge, Massachusetts, January 1975.
 3. Shortliffe, E.H., AXLINE, S.G., Buchanan, B.G. and Cohen, S.N. Design considerations for a program to provide consultation in clinical therapeutics. Proceedings of the San Diego Biomedical Symposium, February 1974, 311-319.
 4. Shortliffe, E.H. and Buchanan, B.G. A model of inexact reasoning in medicine. *Mathematical Biosciences* 23, 351-379, 1975.
 5. Ayd, F.J. Rules for neuroleptic therapy. *International Drug Therapy Newsletter* 9, 33-35 (1974).
 6. Gottschalk, L.A., Hausmann, C. and Brown, J.S. A computerized scoring system for use with content analysis scales. *Comprehensive Psychiatry* 16, 77-90, 1975.
 7. Johnson, J.H., Giannetti, R.A. and Williams, T.A. Real-time psychological assessment and evaluation of psychiatric patients. *Behavioral Research Methods and Instrumentation* 7, 199-200, 1975.
 8. Glueck, B.C. Computers at the Institute of Living. In J.F. Crawford, D.W. Morgan and D. Gianturco (Eds.), *Progress in mental health information systems: Computer applications*. Cambridge, Mass: Ballinger Publishing Company, 1974.
 9. Laska, E.M. The Multi-state information system. In J.F. Crawford, D.W. Morgan and D. Gianturco (Eds.), *Progress in mental health information systems: Computer applications*. Cambridge, Mass: Ballinger Publishing Company, 1974.
 10. Sletten, I.W., and Hedlund, J.L. The Missouri automated Standard System of Psychiatry: Current status, special problems and future plans. In J.F. Crawford, D.W. Morgan and D. Gianturco (Eds.), *Progress in mental health information systems: Computer applications*. Cambridge Mass: Ballinger Publishing Company, 1974.
 11. Spitzer, R.L. and Endicott, J. Can the computer assist physicians in psychiatric diagnosis? *American Journal of Psychiatry*, 131, 523-530, 1974.
 12. Sletten, I.W. and Hedlund, J.L. The future of computers and actuarial methods in mental health practice. Presented at the International College of Psychosomatic Medicine Symposium IV: Rating Devices and Information Processing in Psychosomatics, Catholic University, Rome, Italy, September 16-20, 1975.

APPENDIX A

OVERVIEW OF ARTIFICIAL INTELLIGENCE RESEARCH

B. G. Buchanan and E. A. Feigenbaum
Stanford University

We give here a brief overview of artificial intelligence (AI) taken from a description of the Stanford Artificial Intelligence Laboratory. The articles following the overview are taken from a preliminary draft of a handbook about AI being written at Stanford under Professor Feigenbaum's supervision. The intent of the articles is to convey some sense of the techniques, problems and successes of AI. Only a few of the most relevant articles are reproduced here.

OVERVIEW

Artificial intelligence is the name given to the study of intellectual processes and how computers can be made to perform them. Some workers in the field believe that it will be possible to program computers to carry out many intellectual process now done by humans. However, almost all agree that we are not very close to this goal and that some fundamental discoveries must be made first. Therefore, work in AI includes trying to analyze intelligent behavior into more basic data structures and processes, experiments to determine if processes proposed to solve some class of problems really work, and attempts to apply what we have found so far to practical problems.

The idea of intelligent machines is very old in fiction, but present work dates from the time stored program electronic computers became available starting in 1949. Any behavior that can be carried out by any mechanical device can be represented in a computer, and getting a particular behavior is "just" a matter of writing a program unless the behavior requires special input and output equipment. It is perhaps reasonable to date AI from A.M. Turing's 1950 paper. Newell, Shaw and Simon started their group in 1954 and the M.I.T. Artificial Intelligence Laboratory was started by McCarthy and Minsky in 1958. [The Stanford AI Lab was started in 1963.]

Board Games. Early work in AI included programs to play games like chess, checkers, kalah and go. The success of these programs was related to the extent that human play of these games makes use of mechanisms we didn't understand well enough to program. If the game requires only well understood mechanisms, computers play better than humans. Kalah is such a game. The best rating obtained in tournament play by a chess program so far is around 1700 which is a good amateur level. The chess programmers hope to do better.

Formal Reasoning. Another early problem domain was theorem proving in logic. This is important for two reasons. First, it provides another area in which our accomplishments in artificial intelligence can be

compared with human intelligence. Again the results obtained depend on what intellectual mechanisms the theorem proving requires, but in general the results have not been as good as with game playing. (This is partly because the mathematical logical systems available were designed for proving metatheorems about logic rather than for proving theorems in logic.)

The second reason why theorem proving is important is that logical languages can be used to express what we wish to tell the computer about the world, and we can try to make it reason from this what it should do to solve the problems we give it. It is quite difficult to express what humans know about the world in the present logical languages or in any other way. Some of what we know is readily expressed in natural language, but much basic information about causality and what may happen when an action is taken is not ever explicitly stated in human speech. This gives rise to the representation problem of determining what is known in general about the world and how to express it in a form that can be used by the computer to solve problems.

Publications. The results of current research in artificial intelligence are published in the journal Artificial Intelligence, and in more general computer science publications such as those of the ACM and the British Computer Society. The ACM has a special interest group on artificial intelligence called SIGART which publishes a newsletter. Every two years there is an international conference on artificial intelligence which publishes a proceedings. The fourth and most recent was held in the U.S.S.R. at Tbilisi in the September 1975 and the proceedings are available.

SUMMARY ARTICLES ON SELECTED TOPICS

The following are selected articles on various aspects of Artificial Intelligence research taken from the current collection of articles in the AI handbook effort. A complete outline of the articles planned can be found in Appendix B. The following articles include discussions of production systems, rote learning, speech understanding, and PLANNER.

PRODUCTION SYSTEMSGENERAL DESCRIPTION

A PRODUCTION SYSTEM consists of a set of rules (the productions), a data base and an interpreter for the rules. The data base is a collection of symbols. The interpreter tries to match the left hand side of each production to the data base. The interpreter performs the processes on the right hand side of the production if the condition on the left hand side matches some element in the data base. The productions are generally ordered so that if the condition on the left hand side of more than one production matches an element in the data base, the production higher in the order takes priority.

DATA BASE EXAMPLES

The data base of a production system may be simply a set of symbols intended to reflect the state of the world. Some production systems are intended to model a memory mechanism, for example, "short term memory". For these, each element of the data base may represent some piece of knowledge. Examples of systems modeling short term memory are PSG [Newell, 1973] and VIS [Moran, 1973]. Sample elements from the data base for VIS are

(HEAR NORTH EAST % END)
(L-2 LINE EAST P-2 P-1)

The data bases for knowledge-based experts such as MYCIN [Shortliffe 1975] and DENDRAL [Feigenbaum 1971, Smith 1972] contain facts and assertions about their respective domains of knowledge. For example, the data base in the DENDRAL system contains complex graph structures which represent molecules and molecular fragments. Sample elements from the MYCIN data base are

(IDENTITY ORGANISM-1 E.COLI .8)
(SITE CULTURE-2 BLOOD 1.0)

A third type of data base is the "token stream approach" in which the data base is a linear stream of tokens accessible only in sequence. An attempt is made to match each production to the beginning of the stream and if a match succeeds, characters in the matched segment may be deleted or modified or new characters may be added. This data base organization was used in LISP70 [Tesler 1973].

In all the production systems described above the data base is the only storage medium for all variables of the system. There is no separate control state information such as a program counter or stack as is used in procedurally-oriented languages. The data base is accessible to every rule in the system and thus serves as a communication channel. The contents of the data base always reflect the current state of the production system.

VARIATIONS OF PRODUCTION SYSTEMS

Production systems have been used in many different programs and programming environments. Many variations of production systems are possible due to differences in the ordering and accessing of rules. The productions are themselves a source of variation in production systems. It is possible to match against the right hand side of the productions instead of the left hand side to obtain a recognizer for symbolic strings. It is also possible to view the left hand side as a goal to be achieved by matching the right hand side of the production. The data base may be a source of variation in production systems as has been discussed above.

ENVIRONMENT

Production systems are particularly appropriate in a domain consisting of a large number of independent states requiring independent actions. The states and actions can be modeled easily using rules, which are also modular in nature. Procedure-oriented systems often find it difficult to update and maintain large numbers of state variables. Production systems are particularly appropriate in this instance. Each production can be viewed as a "demon" ready to be invoked when a particular system state occurs. Production systems are also appropriate where the ability to recognize and react to small variations in the domain is important.

REFERENCES

An excellent reference that discusses in detail many aspects of production systems is a paper by R. Davis and J. King entitled "An Overview of Production Systems", (A.I. Memo 271, Stanford Computer Science Department, November 1975). Other references used in this article are:

Minsky M., Computation Finite and Infinite Machines, Prentice-Hall, 1972.

Newell A., Simon H., Human Problem Solving, Prentice-Hall, 1972.

ROTE LEARNING

BRIEF DESCRIPTION AND HISTORY

Rote learning is a technique which effectively increases the depth of tree searches by recognizing nodes (situations) that have been encountered and evaluated previously. This is done by consulting a file which contains for each node previously encountered a description of that node and the result of the evaluation at that encounter.

This technique was first used by A. L. Samuel in his Checker playing program [See Samuel 1959]. The program used rote learning to accumulate experience over the games it played.

ENVIRONMENT

Rote learning is particularly useful when searching game type trees where the value of a position (node, state) is determined by use of an evaluation function.

TECHNIQUE

Assume that there exists a list of nodes which have been evaluated previously. Associated with each node description is an evaluation. This list will be called the memory file. At the very beginning of the learning process, the file contains nothing in the list. The steps below show how the file is built up by the rote learning process.

The basic steps to rote learning are:

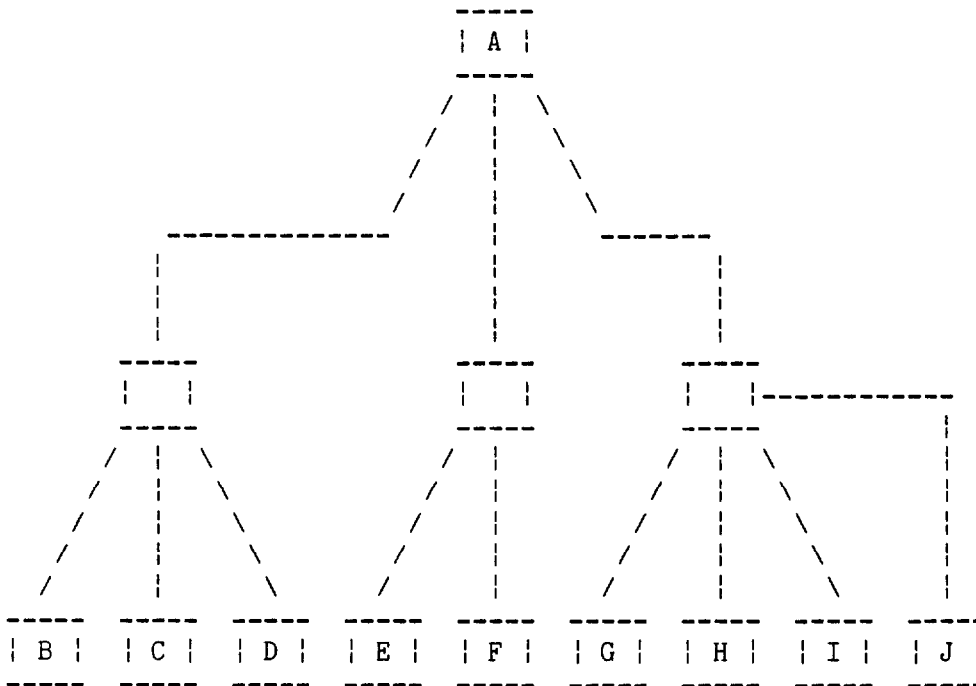
- 1) From the current node which is to be evaluated, form the tree which is to be searched. The form and size of the tree may be governed by a set of heuristics. (For example, expanding the tree fully to a depth of three ply)
- 2) Evaluate the deepest nodes as follows:
 - a) Examine the memory file to see if any of the deepest nodes have been previously evaluated. If so, retrieve from the file the evaluations of these nodes. (Effectively then, these nodes have been evaluated by a further tree search.)
 - b) Any of the deepest nodes not present in the memory file should now be evaluated by the evaluation function.
- 3) Now that all of the deepest nodes have been evaluated back up the tree in the usual min-max fashion to obtain the evaluation for the current node, and to obtain the decision.

- 4) Save a description of the current node and its value in the memory file.

EXAMPLE

Assume that we are playing a game, and that the tree search heuristic is to completely expand the tree to a depth of two ply. Further suppose that we have arrived at a node A, which we wish to evaluate. Assume that this is not our first game, so that the memory file is not empty. We follow the steps of the rote learning procedure as shown:

- STEP 1: From node A, we expand the game tree to a depth of two ply, and label the deepest nodes B through J:



- STEP 2: Looking in the memory file we discover that nodes B, C, E, F, and H have been previously evaluated, so that we already have their values. Thus we apply the evaluation function only to nodes D, G, I, and J to obtain their values.

- STEP 3: Now that the deepest nodes have values, we can go up the tree in the standard fashion, eventually assigning a value to node A, and deciding which branch to take.

- STEP 4: Now that we have a value for node A, we place a

description of the node (for instance, the state vector description) and the value of the node in the memory file for possible future use.

BENEFITS OF ROTE LEARNING

As can be seen from the example, since the values of nodes B, C, E, F, and H were retrieved from the memory file, there is in effect a tree search emanating from each of these nodes, and this tree search took place sometime in the past. Thus the depth of the tree search for node A is only 2 ply in some areas, and of greater ply in others. Now if node A itself is ever retrieved from the memory file for evaluation of another node, the depth of the tree for that node is even greater.

LIMITATIONS

- 1) A potential problem with implementing rote search is the storage and retrieval of information from the memory file, especially as the file grows in size. In cases where rote learning is used on a problem of significant size searching the memory file becomes a task which can take the majority of effort. Techniques from other areas of computer science may be used to aid in efficiently maintaining and searching this file. In addition, it is sometimes necessary or desirable to cull the file (delete entries). In this case, heuristics must be devised to determine which entries to keep and which should be purged.
- 2) It is difficult to use rote learning in conjunction with learning schemes which modify the evaluation function (for example, signature tables). The reason for this is that once the evaluation function is changed, in principle every previously evaluated node in the memory file should be re-evaluated, so that the values of newly evaluated nodes and previously evaluated nodes may be meaningfully compared.

COMMENTS

- 1) Samuel found in his Checker playing program that rote learning worked best in the opening and end games. He hypothesized that rote learning functions reasonably well where the results of any specific action are long delayed, or in situations where highly specialized techniques are required (the Checker playing program learned to avoid obvious traps in the end game).
- 2) By slight modification of the procedure described above rote learning can be used with other true search techniques, such as the alpha-beta search, plausibility ordering, or tree-pruning. The fact to realize is that it is not necessary to expand the tree before looking nodes up in the memory file.
- 3) Samuel observed that rote learning can cause nodes which may both lead to winning situations to receive equal weight in decision making, although one of the nodes may lead to a win much more quickly than the

other. Since it is usually desirable to play a shorter game, the depth of the tree search which leads to each node's evaluation should be considered in this case; the node which has a smaller number of plys to the win should be chosen. Thus it may be necessary to store in the memory file the depth of the search for each node stored in the file.

REFERENCES

Samuel, A.L.; "Some Studies in Machine Learning Using the Game of Checkers," IBM Journal 3, 211-229 (1959). Reprinted (with minor additions and corrections) in COMPUTERS AND THOUGHT, edited by Feigenbaum and Feldman, McGraw-Hill, 1963.

SPEECH UNDERSTANDING

Introduction:

The aim of a "speech understanding" system is determination, for spoken utterances, of the intended message in relation to the accomplishment of some task and in spite of indeterminacies and errors in generation, transmission, and reception of the utterance. This is to be distinguished from the aim of a "speech recognition" system, which is provision of an orthographic transcription of the sounds and words corresponding to the acoustic signal. Thus the aim of a speech understanding system does not necessarily include production of an accurate phonetic transcription of the input signal, or an accurate list of the successive words of the input (although it must surely correctly recognize most of them). In other words, if a situation arises in which acoustic processing is unable to resolve the decision between two phonemes or words at a particular point in an utterance, but the overall system is still able to decide the meaning of the sentence, then the sentence is deemed to have been correctly understood.

It seems apparent that a speech recognition system requires a number of different types of processing, each of which corresponds to a different source of information, in order to achieve its aims. It is now well established that knowledge of vocabulary, syntactic, semantic, and pragmatic constraints of a language is required to compensate for errors and uncertainties in the acoustic realization of an utterance.

In summary, a speech understanding system, as presently conceived, will generally fit the following description.

- 1) The system is organized into a number of levels, starting with the acoustic and working up to the syntactic and semantic.
- 2) Action is generally from the lower levels upward, utilizing programs that incorporate knowledge of each particular level.
- 3) Task limitations are used at several levels to help make selections.
- 4) The higher levels are sometimes used in a feedback mode at lower levels to help make selections.

History:

Speech recognition research has yielded significant results in the case of isolated words (accuracy greater than 95%). The primary emphasis has been on acoustic processing and classical pattern re-cognition and matching techniques. Straight-forward extrapolation of these techniques to continuous speech recognition, however, has not proved successful. It is felt that a major reason for the difficulties encountered is that the information used by humans in understanding speech is not completely contained in the acoustic representation of the speech signal. Experiments by Klatt and Stevens (1972) in the area of spectrogram reading showed that

the performance obtained by human experts for phonetic segmentation and labelling without conscious appeal to syntactic, semantic, and vocabulary constraints was: approximately 75% correctly labelled, 15% mislabelled, and 10% missed. When these other sources of knowledge were used, the success rate for word identification rose to 96%. These results have greatly influenced recent research in speech understanding.

Possible Applications:

Speech would be an appropriate input channel to a computer in many situations. The average output data rate is higher for speech than for writing or typing. Use of the speech channel does not tie up other effectors, such as hands, eyes, feet, or ears. It can therefore be used while in motion or in parallel with other channels. Speech is also a preferred channel for spontaneous communication of the type that is found in an interactive environment.

Long range applications are readily listed. They might include for example, automatic dictation systems, voice-response order takers, or in the computer area, a voice operated graphics terminal.

In the shorter term, several tasks have been suggested as possible vehicles for research in speech understanding (Newell et al 1973). They are:

- 1) Querying a Data Management System
- 2) Data Acquisition of Formatted Information
(voice-key-punch)
- 3) Querying the Operational Status of a Computer
- 4) Consulting on the Operation of a Computer (i.e., a voice-operated HELP)

Unsolved Problems:

The following is a brief discussion of unsolved problems in speech understanding following Newell(1973) and roughly ordered in terms of system level (i.e. from acoustic at the lowest to semantic at the highest).

The essential problem of continuous speech at the acoustic level is phoneme-level identification and not necessarily segmentation between words. There is, however, a significant amount known about acoustic-phonetic and phonological rules which has yet to be fully exploited in production systems. The difficulty of adapting to multiple speakers of different sexes and with different dialects also remains a problem, although it is hoped that proper normalization of acoustic-phonetic and phonological rules will make them speaker-invariant. Two other acoustic-related problems are environmental noise and possible distortions caused by the communications channel (e.g. the telephone channel).

At higher levels there are problems with allowable vocabularies. Present systems attempt to employ vocabularies in which the words are well separated in a feature space. As vocabularies grow, however, or as the choice of words becomes constrained (by a task domain, for example), then the possible errors in matching can be expected to increase. At the syntactic level, it is questionable how much more progress can be achieved without the use of general grammars, as opposed to simple ad hoc grammars. In this regard, the interface between grammars of this type and the phonemic processing level is not yet well understood.

Semantic support is another problem area since many of the interesting applications of speech understanding do not lend themselves to precise semantic formulation. The spontaneity which is a major advantage to speech input works against an understanding system here.

From the hardware point of view, there remain the expected problems of real-time response, processing power, memory size, systems organization, and cost.

In summary, significant progress in speech understanding awaits developments in many areas. It is hoped, however, that many such developments will occur in the next few years.

References:

- A. Newell et al "Speech Understanding Systems", North-Holland, Amsterdam, 1973.
- D.H. Klatt and K.N. Stevens "Sentence Recognition from Visual Examination of Spectrograms and Machine-Aided Lexical Searching", Conference Record, 1972 Conference on Speech Communication and Processing, Newton, Mass., April 1972.

PLANNERCentral Ideas:

Planner is both a problem solving formalism and a programming language. It stresses the importance of goal-orientation, procedural representation of knowledge, pattern directed invocation of procedures and a flexible backtrack-oriented control structure in a problem solver and in a high level programming language.

Technical Description:

Planner was developed as a formalism for problem solving by Hewitt (1972,1973) and a subset of the Planner ideas was implemented by Sussman et al (1973) in a programming language called Micro-Planner.

Planner is primarily oriented towards the accomplishment of goals which can in turn, be broken down into multiple subgoals. A goal in this context can be satisfied by finding a suitable assertion in an associative data base, or by accomplishing a particular task. Multiple goals may be activated at the same time, as might occur, for example in a problem reduction type of problem solver. The attempt to satisfy a goal is analogous to an attempt to prove a theorem. Planner, however, is not strictly a theorem-prover. The differences are mainly due to the types of knowledge which it can manipulate.

The traditional theorem-prover accepts knowledge expressed in declarative form, as in the predicate calculus; that is, as statements of "fact" about some problem domain. Planner, by contrast, is able to deal as well with knowledge expressed in imperative form; that is, knowledge which tells the problem solver how to go about satisfying a subgoal, or how to use a particular assertion. In fact the emphasis in Planner is on the representation of knowledge as procedures. This is based on the view that knowledge about a problem domain is intrinsically bound up with procedures for its use.

The ability to use both types of knowledge leads to what has been called a hierarchical control structure; that is, any procedure (or theorem in Planner notation) can indicate what the theorem-prover is supposed to do as it continues the proof.

Procedures are indexed in an associative data base by the patterns of what they accomplish. Thus, they can be invoked implicitly by searching for a pattern of accomplishment which matches the current goal. This is known as pattern directed invocation of procedures, and is another cornerstone of the Planner philosophy.

The final foundation of Planner is the notion of a backtrack control structure. This allows exploration of tentative hypotheses without loss of the capability to reject the hypotheses and all of their consequences. This is accomplished by remembering decision points (that is, points in the program at which a choice is made) and falling back to them, in order to make alternate choices, if subsequent computation proves unsuccessful.

Example:

The following, somewhat hackneyed, but still illustrative example is described in pseudo Micro-Planner. We will assume that the data base contains the following assertions.

```
(HUMAN TURING)
(HUMAN SOCRATES)
(GREEK SOCRATES)
```

together with the theorem

```
(THCONSE (X) (FALLIBLE $?X)
          (THGOAL (HUMAN $?X)))
```

where the theorem is a consequent theorem which can be read as - if we want to accomplish a goal of the form (FALLIBLE \$?X), then we can do it by accomplishing the goal (HUMAN \$?X).

We now ask the question "is there a fallible Greek?". This can be expressed as

```
(THPROG (X)
         (THGOAL (FALLIBLE $?X) $?T)
         (THGOAL (GREEK $?X))
         (THRETURN $?X))
```

This program uses a linear approach to answering the question; that is, it first attempts to find something fallible, then check that what it has found is Greek. If so, it returns what it has found.

Consider what happens when this program is applied to the data base above. It first finds nothing that is fallible in the list of assertions, and hence tries the theorem, and searches again for something human. It finds (HUMAN TURING) and binds TURING to \$?X. However, an attempt to prove (GREEK TURING) fails. At this point, the backtrack control structure comes into play. The program returns to the last point at which a choice was made; that is, to the point at which TURING was bound to \$?X. This binding is undone and the data base is searched again for something human. This time (HUMAN SOCRATES) is found and SOCRATES is bound to \$?X. An attempt to prove (GREEK SOCRATES) succeeds and SOCRATES is returned as the value of the THPROG.

This example illustrates, albeit superficially, the basic tenets of the Planner formalism as they apply in a programming language. The reader is encouraged to consult the references for the complete details.

References:

- C. Hewitt, "Description and Theoretical Analysis (using schemas) of PLANNER: A Language for Proving Theorems and Manipulating Models in a Robot", Phd Thesis, MIT, Feb., 1971.
- C. Hewitt, "Procedural Embedding of Knowledge in PLANNER", 2nd IJCAI, 1971.

G. J. Sussman, T. Winograd, and E. Charniak, "MICRO-PLANNER Reference Manual", MIT AI Memo 203A, December, 1971.

APPENDIX B
AI HANDBOOK OUTLINE

NOTE:

The following material describes work in progress and planned for publication. It is not to be cited or quoted out of the context of this report without the express permission of Professor E. A. Feigenbaum of Stanford University.

I. INTRODUCTION

A. Intended Audience

This handbook is intended for two kinds of audience; computer science students interested in learning more about artificial intelligence, and engineers in search of techniques and ideas that might prove useful in applications programs.

B. Suggested Style For Articles

The following is a brief checklist that may provide some guidance in writing articles for the handbook. It is, of course, only a suggested list.

- i) Start with 1-2 paragraphs on the central idea or concept of the article. Answer the question "what is the key idea?"
- ii) Give a brief history of the invention of the idea, and its use in A.I.
- iii) Give a more detailed technical description of the idea, its implementations in the past, and the results of any experiments with it. Try to answer the question "How to do it?".
- iv) Make tentative conclusions about the utility and limitations of the idea if appropriate.
- v) Give a list of suitable references.
- vi) Give a small set of pointers to related concepts (general/overview articles, specific applications, etc.)
- vii) When referring in the text of an article to a term which is the subject of another handbook article, surround the term by +'s; e.g. +Production Systems+.

C. Coding Used In This Outline

This outline contains a list of the major areas of artificial intelligence covered in the handbook. At the lowest level, the outline shows article titles either contained or needed. In the case of an article that is needed, the notation NEED[#] follows the proposed focus of the the article, where # is a number in the interval [0,10]. Low numbers indicate little expected difficulty with the article, whereas high numbers indicate a potentially difficult article. For example, an article on a specific system, where only a minimal amount of reading is required would rate approximately 4, whereas an overview article would likely rate 8 or greater. In the case of articles which already exist in the handbook, the notation done[#] is used, where low numbers indicate that the article needs only minor modifications, and high numbers indicate that major modifications are required. For example, repair of typographical errors and wording could be expected to rate 0-2. Correction of errors in the article might rate 3-6, and major rewrites which require considerable reading would likely rate 7-10.

It should be noted that the real difficulty involved in writing an article is highly dependent on the a priori knowledge of its author.

D. A General View of Artificial Intelligence

- | | |
|--|----------|
| Philosophy | NEED [9] |
| This article might address the kinds of questions raised by Turing's article (CAT), Dreyfus's book, the rebuttals, Lighthill's critique, McCarthy's reply, and so on. | |
| Relationship to Society | NEED [8] |
| This might touch on science fiction, popular misconceptions, the Delphi survey, and so on. | |
| History | NEED [9] |
| Perhaps start with Cybernetics, the Dartmouth conference, and so on. See HPS appendix. Also note the major centers, their focus and personalities. Note the role of ARPA funding on the research, the ties to DEC machines and so on. | |
| Conferences and Publications | NEED [6] |
| AI journal, SIGART, SIGCAS, MI books, IJCAI proceedings, CACM, JACM, Cognitive Psychology, some IEEE (Computers, ASSP, SMC), Computational Linguistics, Special interest conferences: robotics, cybernetics, natural language.
Note the tech note unofficial type documents | |

II. HEURISTIC SEARCH

- | | |
|------------------------------|----------|
| A. Heuristic Search Overview | NEED [9] |
|------------------------------|----------|

Algorithmic presentation of "heuristic search" procedure.

Heuristics for choosing promising nodes to expand next,
 heuristics for choosing operators to use to expand a node.

Meta-rules: using heuristics to choose relevant heuristics.

Pervasive character of the combinatorial explosion.

Arguments (both formal and intuitive) supporting the use of
 heuristic search to muffle this explosion.

Formal: Completeness of A*; Knuth's recent work on
 alpha-beta search.

Opportunities for future research

Where do heuristics come from?

(see Simon's current work; meta-rules; meta-meta-...?)

Modifying heuristics based on experiences

(see Berliner's current work)

Working with symbolic, rather than numerical, values for nodes

Coding heuristics as production rules

(e.g.: view Mycin as a heuristic search)

Situations NOT suited to attack by heuristic search

Typically: non-exponential growth process; no search anyway

(e.g., finding roots of a quadratic equation)

Identity problems

Disguising Heuristic Search as something else

Disguising something else to appear to be a Heuristic Search

B. Search Spaces

- | | |
|---|----------|
| 1. Overview | NEED [8] |
| The concept of a search space; how a search space
can be used to solve (some) problems; different
representations, different spaces | |
| 2. State-space representation | done [6] |
| [2 articles exist here, which ought to be unified] | |
| 3. Problem-reduction representation | done [3] |
| 4. AND-OR trees and graphs | done [4] |

C. "Blind" Search Strategies

- | | |
|-----------------------------|----------|
| 1. Overview | NEED [5] |
| 2. Breadth-first searching | done [2] |
| 3. Depth-first searching | done [2] |
| 4. Bi-directional searching | NEED [6] |

discuss heuristics. MI articles by ira Pohl.

- 5. Minimaxing done [3]
- 6. Alpha-Beta searching done [3]

D. Using Heuristics to Improve the Search

- 1. Overview done [7]
 - The idea of a heuristic
 - The idea of a heuristic evaluation function
 - savings in change of representation.
- 2. Best-first searching done [4]
 - (Ordered-search) but need to add: Martelli's work (ask Nils for a draft of this)
 - speech rec: IJCAI-3 (Paxton), Reddy's book
- 3. Hill climbing done [3]
- 4. Means-ends analysis done [3]
- 5. Hierarchical search, planning in abstract spaces NEED [4]
 - Abstrips (Sacerdoti)
- 6. Branch and bound searching done [4]
- 7. Band-width searching NEED [4]
 - Harris - AI journal

E. Programs employing (based on) heuristic search

- 1. Overview NEED [7]
 - Comparison of systems. Results & limitations.
 - (This first article should be written later as an introduction to the following articles.)
- 2. Historically important problem solvers
 - a) GPS NEED [4]
 - b) Strips NEED [4]
 - c) Gelernter's Geom. Program NEED [3]

III. Natural Language

A. Overview

- 1. Early machine translation done [5]

Failures of straight forward approaches

2. History and Development of N.L. NEED [8]
 Main ideas (parsing, representation)
 comparison of different techniques. mention ELIZA, PARRY.
 Include Baseball, Sad Sam, SIR and Student articles here.
 see Winograd's Five Lectures, Simmon's CACM articles.

B. Representation of Meaning
 (see section VII -- HIP)

C. Syntax and Parsing Techniques

1. overviews
 a. formal grammars done [3]
 b. parsing techniques NEED [6]
2. augmented transition nets, Woods done [3]
3. Shrdlu's parser (systemic grammars) done [5]
4. Case Grammars Bruce (AI Journal, 1/76) NEED [5]
5. CHARTS - well formed substrings NEED [6]
6. GSP syntax & parser NEED [6]
7. H. Simon - problem understanding NEED [7]
8. transformational grammars done [5]

D. Famous Natural Language systems

1. SHRDLU, Winograd NEED [5]
2. SCHOLAR NEED [5]
3. SOPHIE NEED [5]

E. Current translation techniques NEED [8]
 Wilks' work, commercial systems (Vauquois)

F. Text Generating systems NEED [8]
 Goldman, Sheldon Klein, Simmons and Sloan (in S&C)

IV. AI Languages

- A. Early list-processing languages done [3]
 overview article

languages like COMIT, IPL, SLIP, SNOBOL, FLPL

Ideas: recursion, list structure,
associative retrieval

B. Language/system features

0. Overview of current list-processing languages NEED [7]

1. Control structures, what languages they are in and examples of their use. NEED [6]

Backtracking (parallel processing)
Demons (pseudo-interrupts)
Pattern directed computation

2. Data Structures (lists, associations, bags, tuples, property lists,...) NEED [5]

Once again, examples of their use is important here.

3. Pattern Matching in AI languages see Bobrow & Raphael NEED [6]

4. Deductive mechanisms see Bobrow & Raphael NEED [5]

C. Current languages/systems

1. LISP, the basic idea done [2]

2. INTERLISP NEED [5]

3. QLISP (mention QA4) done [3]

4. SAIL/LEAP done [2]

5. PLANNER done [2]

6. CONNIVER done [2]

7. SLIP NEED [4]

8. POP-2 NEED [4]

9. SNOBOL NEED [4]

10. QA3/PROLOGUE (see thm. prov.)

V. AUTOMATIC PROGRAMMING

- A. Overview done [7]
- B. Program Specification Techniques
 - i.e. how does the user describe the program to be synthesized?
 - an overview article including various methods NEED[9]
 - see SAFE system (ISI), Green's tech. report, DSSL, Smith's graphic specification, and include general remarks on the high-level language methods
- C. Program Synthesis techniques NEED[9]
 - given a description of the program in some form, generate the actual program
 - 1. Traces done[3]
 - 2. Examples done[3]
 - (include Biggerstaff at U. of Washington)
 - 3. Problem solving applications to AP NEED[9]
 - including classical problem-solving techniques, plan modification, "pushing assertions across goals," and theorem proving techniques. (debugging (Sussmans's Hacker), Simon's Heuristic Compiler, and Prow (Waldinger) & QA3) (Should Theorem-Proving-Techniques remain a separate article?)
 - 4. Codification of Programming Knowledge NEED[?]
 - see C.Green's work, Darlington, Rich & Shrobe
 - 5. Integrated AP Systems NEED[?]
 - see Lenat's original work, Heidorn, Martin's OWL, PSI at SAIL
- D. Program optimization techniques NEED [7]
 - How to turn a rough draft into an efficient program. See Darlington & Burstall, Low, Wegbreit, Kant.
- E. Programmer's aids NEED [7]
 - (Interlisp's DWIM, etc)
- F. Program verification NEED [7]
 - (IJCAI 3)

VI. THEOREM PROVING

- A. Overview NEED [9]
- B. Resolution Theorem Proving
 - 1. Basic resolution method done [4]

2. Syntactic ordering strategies done [2]

3. Semantic & syntactic refinement done [2]

[4. other strategies?]

C. Non-resolution theorem proving

1. Natural deduction done [3]

2. Boyer-Moore

3. LCF

D. Uses of theorem proving

1. Use in question answering done [3]

2. Use in problem solving done [6]

3. Theorem Proving languages
(QA3, Prologue) NEED [5]

4. Man-machine theorem proving
(Bledsoe) NEED [6]

E. Predicate Calculus done [5]

F. Proof checkers

VII. Human Information Processing -- Psychology

(see Perry's outline for details and references)

A. Perception NEED [9]

An overview of relevant work in psychology
on attention, visual and auditory perception,
pattern recognition. Applied perception (PERCEIVER).
Difficulties resulting from inability to introspect.

B. Memory and Learning

1. Basic structures and processes in IPP NEED [9]

Short- and Long-term memory, Rehearsal, Chunking,
Recognition, Retrieval, recall, Inference and
question-answering, Semantic vs. episodic memory,
Interference and forgetting, Type vs. token nodes
Simon - Sciences of the Artificial

2. Overview of memory models, Representation NEED [10]

How to get to the airport: A comparison of the various models.

a. Associative memory models

1. semantic nets NEED [9]
 Quillian (TLC), Nash-Weber (BBN)
 Shapiro, Hendricks (SRI), Wood's article
 in Bobrow & Collins, Simmons (S&C)
2. HAM (Anderson & Bower) NEED [7]
3. LNR: Active Semantic Networks NEED [6]
4. Componential analysis NEED [9]
 Jakendoff, Schank (conceptual
 dependency), (MARGIE), G. Miller
5. EPAM NEED [5]
6. Query languages NEED [7]
 Wood's (1968), Ted Codd (IBM SJ)

b. Other representations

1. Production systems done [1]
2. Frame systems (Minsky, Winograd) done [7]
3. Augmented Transition Networks done [3]
4. Scripts (Schank, Abelson) NEED [7]

C. Psycholinguistics NEED [9]

A prose glossary including:

Competence vs. performance models, Phonology vs. syntax vs. semantics vs. pragmatics, Surface vs. deep structure, Taxonomic grammars, generative grammars, transformational grammars, Phrase-structure rules, transformation rules, Constituents, lexical entries
 Parsing vs. generation, Context-free vs. Context-sensitive grammars, Case systems (e.g., Bruce AI article)

D. Human Problem Solving -- Overview NEED [8]

1. PBG's done [1]
2. Concept formation (Winston) done [2]
3. Human chess problem solving NEED [6]

E. Behavioral Modeling

- | | |
|--|----------|
| 1. Belief Systems
Abelson, McDermott | NEED [8] |
| 2. Conversational Postulates (Grice, TW) | NEED [5] |
| 3. Parry | NEED [5] |

VIII. VISIONA. Overview NEED [9]

This article should discuss the early work in vision; its roots in pattern recognition, character recognition, Pandemonium, Perceptrons and so on. (i.e.. the pre-Roberts work). It should discuss the main ideas of modern vision work as a leadin to the more specific articles, for example the use of hypothesis, model, or expectation driven strategies. It should also discuss the way in which the focus of the field flip-flops from front end considerations to higher level considerations with time.

B. Polyhedral or Blocks World Vision

An overview article should include the major ideas in this work together with brief summaries of the work of the major investigators. In addition, separate articles should be written on the work of those listed below.

- | | |
|---|----------|
| Overview
(Roberts, Huffman and Clowes, Kelley,
Shirai and others listed below) | NEED [7] |
| Guzman | done [2] |
| Falk | NEED [5] |
| Waltz | NEED [7] |
| This article should contain more general material on constraint satisfaction, drawn possibly from Montenari and Fikes | |

This exhausts my list. Please add others or delete some of mine if appropriate.

It has been suggested [Bolles] that the most instructive method of writing these articles would be to provide simple examples of the problems attacked by the various programs.

C. Scene Analysis

Overview NEED [9]

This article should describe or point to detailed strategies used, and the present state of the art.

The following articles should be written or modified to describe the specialized tools of scene analysis. See Duda and Hart.

Template Matching (a non-mathematical description) NEED [5]

Edge Detection done [4]

Homogeneous Coordinates done [7]
This article should be modified to include the general questions of the perspective transformation, camera calibration, and so on.

Line Description done [4]

Noise Removal done [4]

Shape Description done [4]

Region Growing (Yakamovsky, Olander) done [3]

Contour Following NEED [4]

Spatial Filtering NEED [4]

Front End Particulars NEED [6]
This article should contain some description of the methods and effects of compression and quantization for example.

Syntactic Methods NEED [5]

Descriptive Methods NEED [6]
See Duda and Hart, and Winston

D. Robot and Industrial Vision Systems

Overview and State of the Art NEED [9]

Hardware NEED [8]

E. Pattern Recognition

It's not clear just where this discussion should go, or what level of detail is required.

Overview done [8]

This article needs to be refocussed and cleaned up

Statistical Methods and Applications NEED [9]

Descriptive Methods and Applications NEED [8]

F. Miscellaneous

Multisensory Images NEED [7]

Perceptrons NEED [6]

IX. SPEECH UNDERSTANDING SYSTEMS

Overview (include a mention of ac. proc.) done [3]

Integration of Multiple Sources
of Knowledge NEED [9]

For example the blackboard of the HEARSAY II system

HEARSAY I done [4]

HEARSAY II done [5]

SPEECHLIS done [2]

SDC-SRI System (VDMS) NEED [7]

DRAGON done [6]

Jim Baker's original system plus Speedy-Dragon by
Bruce Lowerre. This article is a little harder than
the other system articles because the methods used
may be unfamiliar to some.

X. ROBOTICS

Overview NEED [9]

This article should discuss the central issues and
difficulties of the field, its history, and the
present state of the art.

Robot Planning and Problem Solving NEED [8]

For example, STRIPS and ABSTRIPS. This article
could be quite general depending on the point of
view taken.

Arms NEED [8]

Explain the difficulties of control at the bottom
level, system integration, obstacle avoidance
and so on. Also note the problems with integration
of multi-sensory data, for example vision and

touch feedback.

Present Day Industrial Robots	NEED [7]
Robotics Programming Languages For example WAVE, and AL (a short article)	NEED [6]

XI. Applications of AI

An overview article. What are the attributes of a suitable domain? Custom crafting - theory vs. actual use. (See EAF: 225 notes, 1972) NEED [8]

A. Chemistry

1. Mass spectrometry (DENDRAL, CONGEN, meta-dendral) done [6]
2. Organic Synthesis
Overview NEED [8]
Summarize work of Wipke, Corey, Gelernter, and Sridharan

B. Medicine

1. MYCIN done[1]
2. Summarize DIALOG(Pople), CASNET(Kulikowski), PAUKER'S MIT work, and the Genetics counselling programs NEED[7]

C. Psychology and Psychiatry

Protocol Analysis (Waterman and Newell) NEED [6]

D. Math systems

1. REDUCE NEED [4]
2. MACSYMA (mention SAINT) NEED [6]

E. Business and Management Science Applications

1. Assembly line balancing (Tonge) NEED [5]
2. Electric power distribution systems (MI) NEED [5]

F. Miscellaneous

1. LUNAR NEED [5]
2. Education NEED [7]
Papert, or more ?

3. SRI computer-based consultation NEED [6]

4. RAND--RITA production rule system for
intelligent interface software NEED [5]

I. Miscellaneous

Overview of music composition and aesthetics done [7]

XII. Where do these go?

Reasoning by analogy done [4]

Intelligence augmentation done [5]

Chess done [5]

XIII. Learning and Inductive Inference

Overview NEED [9]

Samuel Checker program NEED [5]

Winston done [2]

Pattern extrapolation problems--Simon,
Overview of Induction NEED [5]

APPENDIX C

HEURISTIC PROGRAMMING PROJECT WORKSHOP

In the first week of January 1976, about fifty representatives of local SUMEX-AIM projects convened at Stanford for four days to explore common interests. Six projects at various degrees of development were discussed during the conference. They included the DENDRAL and META-DENDRAL projects, the MYCIN project, the Automated-Mathematician project, the Xray-Crystallography project, and the MOLGEN project. Because of the interdisciplinary nature of each of these projects, the first day of the conference was reserved for tutorials and broad overviews. The domain-specific background information for each of the projects was presented and discussed so that more technical discussions could be given on the following days. In addition the scope and organization of each of the projects was presented focusing on the tasks that were being automated, how people perform these tasks, and why the automation was useful or interesting.

In the following days of the workshop, common themes in the management and design of large systems were explored. These included the modular representations of knowledge, gathering of large quantities of expert knowledge, and program interaction with experts in dealing with the knowledge base. Several of the projects were faced with the difficulties of representing diverse kinds of information and with utilizing information from diverse sources in proceeding towards a computational goal. Parallel developments within several of the projects were explored, for example, in the representation of molecular structures and in the development of experimental plans in the MOLGEN and DENDRAL projects. The use of heuristic search in large, complex spaces was a basic theme to most of the projects. The use of modularized knowledge typically in the form of rules was explored for several of the projects with a view towards automatic acquisition, theory formation, and program explanation systems.

For each of the projects, one session was devoted to plans for future development. One of the interesting questions for these sessions was the effect of emerging technology on feasibility of new aspects of the projects. The potential uses of distributed computing and parallel processing in the various projects were explored, particularly in the context of the DENDRAL project.

Most of the participants felt that the conference gave them a better understanding of related projects. And because many members of the SUMEX-AIM staff actively participated, the workshop also provided all projects with information about system developments and plans. The discussions and sharing of ideas encouraged by this conference has continued through a series of weekly lunches open to this whole community.

Number	1
Average Delay	365.0
Std Deviation	.0
Minimum Delay	365
Maximum Delay	365

1011 WASHINGTON D.C. WASSR1 C ** 703/841-9560

July 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number		1		
Average Delay		204.0		
Std Deviation		.0		
Minimum Delay		204		
Maximum Delay		204		

September 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number		5		
Average Delay		177.6		
Std Deviation		38.9		
Minimum Delay		123		
Maximum Delay		227		

October 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	1			
Average Delay	153.0			
Std Deviation	.0			
Minimum Delay	153			
Maximum Delay	153			

November 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	2			
Average Delay	144.5			
Std Deviation	13.5			
Minimum Delay	131			
Maximum Delay	158			

1012 CHICAGO ILLINOIS CHI2 E ** 312/346-4961

December 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	1	3		
Average Delay	346.0	393.0		
Std Deviation	.0	160.8		
Minimum Delay	346	214		
Maximum Delay	346	604		

March 1976

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
--	-------------	-------------	-------------	-------------

Number	2
Average Delay	251.5
Std Deviation	66.5
Minimum Delay	185
Maximum Delay	318

1014 MIDLAND TEXAS MDL1 C 915/683-5645

June 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	2			1
Average Delay	525.0			310.0
Std Deviation	158.0			.0
Minimum Delay	367			310
Maximum Delay	683			310

1017 PALO ALTO CALIFORNIA PA1 C ** 415/494-3900

July 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number			1	
Average Delay			414.0	
Std Deviation			.0	
Minimum Delay			414	
Maximum Delay			414	

1022 WASHINGTON D.C. WAS2 E ** 703/521-6520

July 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number		3		
Average Delay		188.0		
Std Deviation		10.6		
Minimum Delay		173		
Maximum Delay		196		

September 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number		3		
Average Delay		197.3		
Std Deviation		23.5		
Minimum Delay		165		
Maximum Delay		220		

October 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number		2		
Average Delay		261.0		
Std Deviation		3.0		
Minimum Delay		258		
Maximum Delay		264		

November 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	3	12		
Average Delay	242.7	300.4		
Std Deviation	64.5	161.8		
Minimum Delay	153	129		
Maximum Delay	302	774		

December 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number		2		
Average Delay		208.0		
Std Deviation		49.0		
Minimum Delay		159		
Maximum Delay		257		

1023 SEATTLE WASHINGTON SEA1 C 206/622-7930

September 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	1	1		
Average Delay	385.0	391.0		
Std Deviation	.0	.0		
Minimum Delay	385	391		
Maximum Delay	385	391		

March 1976

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number		1		
Average Delay		805.0		
Std Deviation		.0		
Minimum Delay		805		
Maximum Delay		805		

1027 LOS ANGELES CALIFORNIA LA2 E ** 213/683-0451

December 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number		2		
Average Delay		162.0		
Std Deviation		6.0		
Minimum Delay		156		
Maximum Delay		168		

January 1976

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	3			
Average Delay	172.3			
Std Deviation	9.4			
Minimum Delay	161			
Maximum Delay	184			

1034 NEW YORK NEW YORK NYCSR1 E ** 212/532-7615
NEW YORK NEW YORK NYCSR1 E ** 212/551-9322

June 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number		8		
Average Delay		561.9		
Std Deviation		98.9		
Minimum Delay		407		
Maximum Delay		709		

July 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	3	7		
Average Delay	511.3	518.3		
Std Deviation	53.8	105.3		
Minimum Delay	458	407		
Maximum Delay	585	732		

September 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	2	15		
Average Delay	418.0	365.9		
Std Deviation	95.0	187.7		
Minimum Delay	323	187		
Maximum Delay	513	828		

October 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	13	15		
Average Delay	712.2	394.3		
Std Deviation	523.5	147.2		
Minimum Delay	335	182		
Maximum Delay	1783	768		

November 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	19	21		
Average Delay	635.4	380.6		
Std Deviation	511.0	55.4		
Minimum Delay	224	264		
Maximum Delay	2183	510		

December 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	13	33		
Average Delay	855.2	931.2		
Std Deviation	996.8	908.4		
Minimum Delay	190	223		
Maximum Delay	2763	3035		

January 1976

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	4	11		
Average Delay	466.0	591.4		

Std Deviation	152.7	180.0
Minimum Delay	226	233
Maximum Delay	621	901

February 1976

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	2	11		
Average Delay	508.5	709.7		
Std Deviation	53.5	160.3		
Minimum Delay	455	466		
Maximum Delay	562	1028		

March 1976

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	8	5		
Average Delay	849.8	581.8		
Std Deviation	315.1	230.1		
Minimum Delay	487	331		
Maximum Delay	1351	953		

April 1976

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	13	6		
Average Delay	1180.4	794.3		
Std Deviation	511.8	304.0		
Minimum Delay	529	471		
Maximum Delay	2108	1346		

1036 NEW YORKNEW YORKNY1E ** 212/344-7445

June 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	4	3		
Average Delay	687.8	495.3		
Std Deviation	66.9	134.8		
Minimum Delay	609	339		
Maximum Delay	756	668		

July 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	6	1		
Average Delay	426.5	847.0		
Std Deviation	77.2	.0		
Minimum Delay	338	847		
Maximum Delay	562	847		

September 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number		4		
Average Delay		380.8		
Std Deviation		34.0		
Minimum Delay		346		
Maximum Delay		428		

1037 LOS ANGELES CALIFORNIA LASR1 C ** 213/629-1561

December 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number		1		
Average Delay		121.0		
Std Deviation		.0		
Minimum Delay		121		
Maximum Delay		121		

1043 ST LOUIS MISSOURI SL1 C 314/421-5110

June 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	12	27	2	
Average Delay	800.4	766.9	309.0	
Std Deviation	211.1	212.4	39.0	
Minimum Delay	431	480	270	
Maximum Delay	1124	1347	348	

July 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	16	83	11	
Average Delay	649.3	679.9	325.9	
Std Deviation	152.9	238.7	53.9	
Minimum Delay	435	243	244	
Maximum Delay	971	1550	420	

August 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	8	27	1	
Average Delay	660.6	601.9	302.0	
Std Deviation	235.8	209.8	.0	
Minimum Delay	242	268	302	
Maximum Delay	942	1079	302	

September 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	8	20	2	
Average Delay	569.4	538.7	369.0	
Std Deviation	221.0	228.4	95.0	
Minimum Delay	333	238	274	
Maximum Delay	988	939	464	

October 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	11	26	2	
Average Delay	517.6	516.3	218.0	
Std Deviation	110.6	168.8	9.0	
Minimum Delay	380	237	209	
Maximum Delay	757	960	227	

November 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number				

Number	2	9	1	1
Average Delay	500.5	532.1	258.0	225.0
Std Deviation	85.5	119.7	.0	.0
Minimum Delay	415	320	258	225
Maximum Delay	586	770	258	225

December 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	4	9	1	
Average Delay	498.0	345.9	294.0	
Std Deviation	157.2	178.6	.0	
Minimum Delay	315	155	294	
Maximum Delay	749	807	294	

January 1976

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	1	14		
Average Delay	374.0	399.6		
Std Deviation	.0	174.1		
Minimum Delay	374	177		
Maximum Delay	374	943		

February 1976

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number		11	3	
Average Delay		344.3	172.0	
Std Deviation		87.9	7.0	
Minimum Delay		153	163	
Maximum Delay		491	180	

March 1976

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	5	12	4	1
Average Delay	849.6	432.7	381.3	160.0
Std Deviation	722.3	265.5	306.2	.0
Minimum Delay	210	238	160	160
Maximum Delay	1779	1200	909	160

April 1976

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	4	10	1	
Average Delay	300.0	279.5	175.0	
Std Deviation	36.0	82.0	.0	
Minimum Delay	251	201	175	
Maximum Delay	347	431	175	

1051 PORTLANDOREGONPOR1 C503/224-0750

August 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number		1		
Average Delay		299.0		
Std Deviation		.0		
Minimum Delay		299		

Maximum Delay 299

December 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	3			3
Average Delay	666.0			229.7
Std Deviation	110.7			14.4
Minimum Delay	519			210
Maximum Delay	786			244

January 1976

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number			4	
Average Delay			458.3	
Std Deviation			154.5	
Minimum Delay			266	
Maximum Delay			614	

1054 SAN JOSE CALIFORNIA CRP2 C ** 408/446-4850

August 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number		1		
Average Delay		211.0		
Std Deviation		.0		
Minimum Delay		211		
Maximum Delay		211		

1060 MOUNTAIN VIEW CALIFORNIA AME1 E ** 415/965-8815

June 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number			3	
Average Delay			287.0	
Std Deviation			88.0	
Minimum Delay			171	
Maximum Delay			384	

July 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number		3		
Average Delay		318.0		
Std Deviation		124.7		
Minimum Delay		220		
Maximum Delay		494		

1063 PITTSBURGHPENNSYLVANIAPIT1 C412/765-3511

June 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number		2		
Average Delay		471.5		
Std Deviation		45.5		
Minimum Delay		426		
Maximum Delay		517		

September 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number		3		
Average Delay		268.7		
Std Deviation		49.5		
Minimum Delay		200		
Maximum Delay		315		

November 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	1			
Average Delay	283.0			
Std Deviation	.0			
Minimum Delay	283			
Maximum Delay	283			

December 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	1			
Average Delay	267.0			
Std Deviation	.0			
Minimum Delay	267			
Maximum Delay	267			

February 1976

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number		1		
Average Delay		668.0		
Std Deviation		.0		
Minimum Delay		668		
Maximum Delay		668		

March 1976

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	1			1
Average Delay	297.0			266.0
Std Deviation	.0			.0
Minimum Delay	297			266
Maximum Delay	297			266

1072 PALO ALTOCALIFORNIAPCOSR1 E ** 415/326-7015

August 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number		1		1
Average Delay		169.0		148.0
Std Deviation		.0		.0
Minimum Delay		169		148
Maximum Delay		169		148

1073 UNIONNEW JERSEYUNISR1 E ** 201/964-3801

June 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number			2	
Average Delay			371.0	
Std Deviation			9.0	
Minimum Delay			362	
Maximum Delay			380	

August 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	1	1		
Average Delay	484.0	692.0		
Std Deviation	.0	.0		
Minimum Delay	484	692		
Maximum Delay	484	692		

October 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	2	1		
Average Delay	769.5	485.0		
Std Deviation	97.5	.0		
Minimum Delay	672	485		
Maximum Delay	867	485		

November 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	7	10		
Average Delay	641.6	689.8		
Std Deviation	204.4	178.2		
Minimum Delay	419	476		
Maximum Delay	1106	1055		

January 1976

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	1			
Average Delay	281.0			
Std Deviation	.0			
Minimum Delay	281			
Maximum Delay	281			

March 1976

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number				

Number	2
Average Delay	688.5
Std Deviation	221.5
Minimum Delay	467
Maximum Delay	910

April 1976

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	1			
Average Delay	1125.0			
Std Deviation	.0			
Minimum Delay	1125			
Maximum Delay	1125			

1112	NEW YORK	NEW YORK	NYCSR2 C ** 212/750-9433
	<u>NEW YORK</u>	<u>NEW YORK</u>	<u>NYCSR2 C ** 212/750-9445</u>

June 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	4	13		
Average Delay	668.5	308.1		
Std Deviation	207.6	51.3		
Minimum Delay	458	232		
Maximum Delay	960	439		

July 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	5	7		
Average Delay	655.2	532.9		
Std Deviation	176.9	104.2		
Minimum Delay	401	356		
Maximum Delay	891	679		

August 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number		1		
Average Delay		600.0		
Std Deviation		.0		
Minimum Delay		600		
Maximum Delay		600		

December 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	1			
Average Delay	894.0			
Std Deviation	.0			
Minimum Delay	894			
Maximum Delay	894			

1116 CHICAGOILLINOISCHISR1 C ** 312/368-4607

August 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number		1		
Average Delay		166.0		
Std Deviation		.0		
Minimum Delay		166		
Maximum Delay		166		

1173 VALLEYFORGEPENNSYLVANIAVFOSR1 E215/666-9190

December 1975

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number	1	4		
Average Delay	311.0	392.8		
Std Deviation	.0	102.4		
Minimum Delay	311	266		
Maximum Delay	311	511		

January 1976

	05:00-09:00	09:00-17:00	17:00-22:00	22:00-05:00
Number		4		
Average Delay		457.5		
Std Deviation		28.2		
Minimum Delay		421		
Maximum Delay		496		

APPENDIX E

MAINSAIL DESIGN SUMMARY

A MACHINE-INDEPENDENT PROGRAMMING SYSTEM

Clark R. Wilcox
SUMEX Computer Project, Stanford University
Stanford, California

ABSTRACT

A general-purpose programming system is being developed for the support of portable software, and as a tool for research into machine-independent code generation. The issues involved in such a design project are discussed, and an overview is given of the approach taken for MAINSAIL.

INTRODUCTION

Much effort is now expended in the development of software whose conceptual framework, at least, is already well-understood and documented. A significant amount of time spent in such development is invariably attributable to the particular environment in which the program will execute, rather than the function of the program itself. An algorithm is easily overwhelmed by implementation details, and its intention obscured by the resulting program. The source language, the operating system, the size of the machine, the file system, the debugging facilities, the time schedule, the demands of efficiency: all seem to conspire against clarity and generality. The original purpose, and the means used to obtain a running program, can become inextricably enmeshed, the result having no application beyond its limited context. The program becomes tied to the machine, the operating system, a particular version of the operating system, and the various local enhancements, and certain terminals, with given keyboards and character sets; it continually becomes obsolete, never works quite right, and dies a certain death when the author departs. And yet essentially the same program is developed for other machines, and meets the same fate. There seems to be neither the time nor the tools to do it right once, and distribute it; indeed, everyone is busy writing his own version.

If a program is to find general use beyond the confines of a particular implementation, the multitude of machine-dependent traps must be defended against at every turn. Whether this necessarily entails a loss in efficiency (program size and execution time), and the inability to use

local features which might otherwise enhance performance, is becoming less clear, and certainly less important as memory and processing rates increase. The programming task is being given increased scrutiny, with an eye to the elimination of duplication, obscurity and inflexibility solely for the purpose of execution-time efficiency. Software is viewed more as a product with general applicability than as a means to an end. The tremendous effort required for a quality software product is resulting in a less tolerant attitude towards programs which must be totally rewritten if "moved" to a new machine.

If programmers had access to programming systems which aided in the creation of portable software, then perhaps we would be surprised at the tasks now considered machine-dependent which could be cast in a more general mold, passed from one machine to another, with possibly minor changes isolated and well-documented. To gain acceptance, such a system must balance several conflicting requirements without adversely affecting its ease of use.

PORTABILITY

The programming system itself must be transportable among a wide variety of machines. Its design must incorporate the means to insure compatible versions among machines, and to allow a new machine to be implemented with a minimum of effort. A language standard, presumably enforced in all implementations, is not sufficient. There is little chance that every version will be totally compatible. A standard retards the introduction of improvements and new ideas, since every implementation requires concurrent upgrading to preserve compatibility. The orchestration of such updates across a broad class of computers is prohibitive. Thus the parallel development of the programming system on many machines is not sufficient, and is an example of the very redundancy which a machine-independent programming system can alleviate. Such is the case with many languages which are now used for program portability, for example FORTRAN, COBOL, BASIC and SIMULA.

If a single version of the system could be written and distributed to all sites, then an elegant solution would be provided to the problem of maintaining compatibility, and hence portability. There would be no need for a language standard, since each site would use the same compiler. Every version would without question be compatible, since there would be only one version. Any changes to the system would be immediately transmitted to all sites by merely sending copies of the updated software. Errors found by one site result in fixes for every site.

This type of distribution can take place if the programming system is written in its own language. All software comprising the MAINSAIL programming system is itself written in the MAINSAIL language. The compiler can compile itself, and its own runtime system. It is easily bootstrapped since it is written in a subset of MAINSAIL which can be compiled by an existing compiler for the language SAIL, from which MAINSAIL is derived. Furthermore, the creation of a MAINSAIL system for a new computer is largely automated by a compiler-generator program.

The programming system itself is one example of the portability of programs written for the system. As a corollary, user programs can be written which will execute correctly on any implementation. The consequences of being able to move programs freely among several computers and operating systems are far-reaching. Programs may be shared among all sites, regardless of what computers are involved. At a single site, the same language can be used on all computers, thus promoting program interchange, and removing the problems involved with using different languages on each computer. If one computer system becomes unavailable, programs may be moved to another. The introduction of new computers may take place without fear that existing programs will become obsolete: it is only necessary that the programming system be implemented on the new system.

EFFICIENCY

In order to compete successfully with existing programming systems, a machine-independent system must offer advantages greater than the penalties derived from its lack of intimacy with the host machine. While this statement is nearly tautological, it nevertheless suggests the tradeoffs between efficiency and portability which must be dealt with in the design of such a system. Machine-independence is more a question of degree than possibility, since, in theory at least, even an extremely limited machine can be made to simulate the operations of the most powerful.

In order to obtain an acceptable level of efficiency, few assumptions concerning the target machines should be embodied in the programming system. It would be unacceptable to model all target machines as stack machines, if this model must be carried to the point of code generation. Similarly, register usage, linkage conventions, addressability, and storage allocation must not be given rigid characteristics if the system is to be truly portable. Interpreted code cannot be emitted in every case. Such considerations seem to rule out the effectiveness of a well-defined "abstract machine" for which code is generated. Instead, the code should be made to fit each target machine as well as most compilers now fit the machines for which they were designed. In many cases MAINSAIL is able to generate better code than existing compilers. For example MAINSAIL produces about 10 percent less code than the SAIL compiler, which was designed for a particular machine (PDP-10).

MACHINE-DEPENDENCIES

Somewhat paradoxically, a machine-independent programming system can benefit from features which support its use in machine-dependent applications. If the language attempts to ban any constructs which it considers machine-dependent, then programs which by their nature are heavily dependent on a particular machine configuration cannot be written. Programmers who would prefer use of the language must turn to another for such purposes; their preferences may be similarly turned.

At the very least, linkage should be allowed to external procedures

written in other languages, so that a library of procedures of local interest can be constructed. If such a procedure is very short, say merely a call to the operating system, then the overhead for a procedure call may be unacceptable. In this case, the ability to insert assembly language directly into the program is most useful.

By its very design, MAINSAIL can benefit from machine-dependencies. Though most of the runtime system is written for portability in MAINSAIL, some system procedures are too machine-dependent to be written once for all computers. When writing these procedures for a particular implementation, it is desirable to use MAINSAIL if possible, because of the ease with which the machine-dependent portion can be interfaced with the machine-independent parts. Thus the entire runtime system may be written in MAINSAIL, which seems almost magical considering that everything else is also written in MAINSAIL.

There is of course a danger in explicitly allowing the introduction of machine-dependencies into the language. Programmers may begin using such constructs when not really necessary, so that the advantages of using a portable language are lost.

LANGUAGE DESIGN

In designing a general-purpose language for portability, one is immediately faced with the problem of data representation, for this is most closely dictated by the underlying machines. The selection of primitive data types must not be too narrow to prevent the full use of more powerful machines, nor too broad to require extensive simulation on smaller machines. Two basic approaches for data definition suggest themselves: offer standard definitions from which the programmer must choose; or give the programmer control over data characteristics such as range and precision. These approaches can be contrasted for the primitive data type integer.

The first would offer one or more standard ranges, for example INTEGER and LONG INTEGER, with ranges corresponding to, say, 16 bits, and greater than 16 bits (an upper bound would be of dubious value). These ranges would correspond to the minimal ranges expected for all computers to be implemented, and the programmer would understand that in a program written for portability, LONG INTEGER would preclude its use on computers with a small word size, unless this type were simulated. On larger computers, INTEGER might be represented with, say, 32 bits, and programs written specifically for such machines could make use of the full range.

The second approach would include, with each declaration, range information, for example the smallest and largest values. The compiler would use this information to allocate the integer, presumably choosing different representations for different ranges. The programmer need consider only the characteristics of his data, rather than the various machines which are to support his program. The inclusion of a range specification is also a useful form of program documentation, and aids the compiler in checking that the variable is properly used. Of course, the programmer must realize the consequences if his integer range is beyond that of a 16-bit word.

MAINSAIL presently offers the first approach with data types INTEGER, LONG (integer), REAL, and DOUBLE (real). LONG and DOUBLE are useful if the hardware provides these extended data types, or they are necessary for the intended applications, but must be supported by software. In the latter case they are expensive to use, and the single precision types should be employed where possible. In either case, machine-dependent considerations are involved in deciding to use these types, and thus they cannot appear in "portable" programs. This approach simplifies the compiler design, and perhaps results in more efficient code for smaller machines, where this is most crucial. The type BITS, for logical operations on bit vectors, is also offered, and defined as providing at least 16 bits. Thus the data types are optimized for ease of implementation, rather than optimal use of storage on machines with larger words. The compiler is never concerned with an attempt to "pack" a data type into the available words.

MAINSAIL says nothing about the bit patterns used to represent data. For example, integers can be represented as ones complement, twos complement, or even decimal. Bit operations are allowed only on the type BITS, with standard conversions among BITS and INTEGER. An INTEGER is converted to BITS by forming the binary representation of the integer (undefined if the integer is negative). Similarly, a BITS is converted to INTEGER by forming the non-negative integer whose binary representation is given by the bits. Thus it can be determined whether a positive integer is odd by converting to BITS and testing the low-order bit, no matter what representation is being used.

Another issue of data representation is the character codes. MAINSAIL offers the type STRING, which is a variable-length sequence of characters (the number of characters is automatically kept track of). There are two operations which are concerned with character codes: the first character of a string may be converted to its integer code; and an integer may be converted to a string of one character. The codes used to store characters within strings are of no consequence; there is only a need for a standard code during the two operations. MAINSAIL decrees that the ASCII codes are in effect whenever an integer is deemed to be a character code. Each implementation is responsible for any necessary conversions to and from the internal codes used in string storage.

In order to allow the runtime system to be largely written in MAINSAIL, some assumptions concerning memory and addressability are necessary. The amount of memory required by each data type is measured in "storage units." The physical interpretation of a storage unit is machine-dependent; for example, a storage unit may be a "byte" or a "word." The number of storage units required by n consecutive values of the same type, for example elements of an array, is n times the size of a single value. However, sizes of consecutive values of differing types cannot be added to obtain a total size, since machine-dependent "padding" may occur between the allocations for alignment purposes.

The type ADDRESS is introduced for manipulating memory addresses. A memory model is adopted which specifies only those addressing characteristics necessary for the simplest memory accesses. For example, an address is not used to indicate a particular character of a string,

since this is not possible on some machines without additional information concerning the location of the character within a word. Associated with each STRING is a "string descriptor" which contains the current length, and the location of the first character. A string descriptor is a primitive data type, since an integer-address pair may not be sufficient.

Addressability, and the associated issue of program linkage, is an area which requires special attention. MAINSAIL allows programs to be written as separate texts, called "segments." These segments are separately compiled, and linked together to form a program in some machine-dependent manner. Inter-segment communication is provided by global data and procedures. Each segment is given a name and characteristics such as MAIN and OVERLAY. A variable or procedure is declared "external" by preceding its declaration with the name of the segment which contains its "internal" occurrence. If a procedure is internal to an OVERLAY segment, then that segment must be brought into memory before the procedure can begin execution. MAINSAIL does not provide the facilities for such overlay handling, but does include the syntax for specifying which segments are overlays.

A machine must provide for an address composed of a static or dynamic base (possibly external), with a static or dynamic offset. Static means that the value does not change during program execution, i.e. it is known at compile-time (within relocation). Thus a computer which does not provide indexing will produce inefficient code. A single level of indirect addressing can also improve the code quality. For example, if an address variable is in memory, it is useful to be able to access, say, an integer pointed to by the address, without first loading the address into an index register.

The syntax of expressions and statements is more distant from the underlying machine, so that there are few difficulties in removing machine-dependencies. Perhaps the overall result is a clear and straightforward syntax, since the prejudices and peculiarities exhibited by more machine-dependent languages are missing. There are no exotic data operations, since every machine would have to support such operations. Probably no machine will have instructions corresponding to every operation, though some come rather close. For example, BITS can be shifted left or right by any amount. Some machines have instructions which do just this; others require several instructions, or even a procedure call. STRING operations are generally too complicated to be carried out in-line, and thus there is no requirement for byte addressability or compact byte-manipulation instructions.

COMPILER DESIGN

The primary consideration in the design of a machine-independent compiler is the interface between what is known about the language and assumed about all target machines, and what is left to be supplied for each implementation. If too much is assumed, then the class of machines is unduly restricted, and clumsy devices may be necessary to resolve a distorted model to reality, resulting in needless inefficiencies. If too little, then the generation of a new system could be a major undertaking, retarding the spread of the system to new machines.

In contrast to a compiler-compiler which has no knowledge of the source language, the MAINSAIL language and compiler evolved by an iterative process. Features which were felt necessary for an efficient compiler were simply put into the language. Similarly, the language was modified in those areas requiring an inordinate amount of time or space for compilation. With regard to optimizations, this intertwining of design may result in additional statements in the compiler, yet a smaller compiler when the optimized version compiles itself.

The compiler consists of two passes in order to cleanly separate the machine-independent and dependent phases. The first pass converts the source program to an intermediate language, and the second translates this intermediate language to the target assembly language (which must be assembled by some machine-dependent assembler not provided by MAINSAIL). The intermediate language consists of operators with a variable number of operands. The operators reflect either MAINSAIL operations, such as addition; program structure, such as procedure entry; or internal information, such as the handling of temporaries. In most cases an operand is a pointer into the symbol table.

This is quite different from an attempt to generate intermediate code for an abstract machine. For example, the intermediate code for "a := a + b" might be <push a>, <add b>, <pop a> if the abstract machine were stack-oriented, whereas MAINSAIL generates <add b a>. In the former case, a register-oriented machine could certainly simulate the pushes and pops, but the generated code would be of dubious quality. A machine with a memory-to-memory add would suffer even more. MAINSAIL, however, generates intermediate code which captures only what is in the source program, with no assumptions concerning the target machine. The <add b a> can involve registers, a stack, memory-to-memory, or even a procedure call.

The second pass consists of a machine-independent part, and a machine-dependent part which is translated from a code-generation language. The machine-independent part is responsible for creating a convenient interface to the machine-dependent part, consistent with the separation between the two. It fetches the intermediate instructions, and sets up the operator and operands for easy accessibility. It supplies answers to questions concerning the operands, or the current code generation environment which it is responsible for maintaining.

MAINSAIL employs a general notion of register which is useful in a number of contexts. An operand is always associated with a memory location, and may be temporarily marked as loaded in a register. The compiler provides several services related to registers, such as: mark an operand in a register, clear a register, or find the "best" free register. It will automatically load and store registers when necessary. A register may also be marked as containing the address of an operand.

The services provided for registers are never invoked unless the code generators either directly request a service, or indicate that registers are to be used in certain situations (for example, to pass procedure parameters). Thus code can be generated for machines with no registers, for example a stack machine (actually, the top of the stack can be modeled as a register). A code-generation environment is created and

maintained which is flexible enough to be of use for a wide variety of computer architectures. Many checks insure the internal consistency of the environment, for example a register cannot be marked with two operands at the same time. By knowing the rules of this environment, code generators can be written for a new computer with minimal effort.

The code-generation language provides a powerful and convenient setting in which to specify code sequences. Declarations give semantic information concerning register usage, storage units, additional symbol table entries, and various parameters used within the compiler and runtime system. A code generator must be written for each intermediate instruction. A generator has available to it services such as those discussed above, and the operands of the intermediate instruction. In general a code-generator looks like the assembly language which it is to produce, except it contains keywords which are replaced during code generation with operand names, registers, or constants. The code-generation language is translated to MAINSAIL, and hence the full power of MAINSAIL is available. In practice, the constructs provided are sufficient for almost all situations which arise during code generation. A code generator usually takes the form of a series of conditions, each followed by pseudo assembly language which is to be processed if the condition is satisfied. The complexity of the conditions is determined by the degree to which the target machine conforms to the general framework provided for code generation, and the amount of optimization desired. Procedures can be used for commonly occurring code sequences.

Since code generators are associated with intermediate instructions, they provide only for local optimization. Because of the extreme ease with which the code generators can be altered, a compiler can be created from the current generators, and its output examined for errors and inefficiencies. Based on this, the generators can be altered, a new compiler created, and so forth. This process continues until the code appears correct, and is sufficiently efficient. Construction of a new compiler from a few changes in the generators can be done in a matter of minutes. Thus a single session spent tuning the generators can produce significant results.

The formal separation of target-machine semantics from the more general aspects of code generation has an exciting potential for research into the design of instruction sets. Since a wide variety of computers can be described with the code generators, experiments can be conducted to test features such as the number of registers, the utility of indirection, or various procedure linkages. Existing machines can be compared to determine which is best suited for a high-level language implementation. For example, an instruction set which allows complete addresses can be compared with one which offers a base with small displacement, to determine which requires the fewest memory accesses. A micro-coded instruction set based on the MAINSAIL intermediate instructions would produce optimized code sequences.

The facility with which code generators can be written makes MAINSAIL accessible to one-of-a-kind machines. For example, there is now under construction a three-address parallel processor with no registers which will use MAINSAIL as its high-level language. Programs can be

written, and the code examined, before the machine is complete (even the assembler for the new machine can be written in MAINSAIL!). Providing such a machine with a high-level language would be a major undertaking if the compiler, runtime system and assembler had to be written in assembly language.

RUNTIME DESIGN

The runtime system provides support during program execution: program initialization, file manipulation, i/o, conversions among string and numeric-bits, string handling, mathematical routines, string and record collection, and dynamic memory allocation. If MAINSAIL is to be used as an implementation language, then it may be desired to limit the size of the runtime package. Since the system procedures are used only in response to implicit or explicit requests, programs may be written which require little, if any, support. For example, programs which involve only arithmetic, logical and address operations, with no i/o, string handling or dynamic storage allocation, may be compiled into assembly language programs which call only the system initialization procedure. By removing this call, a self-sufficient program is obtained which can be combined with hand-coded assembly-language modules. In this sense, MAINSAIL can be regarded as a convenient means of generating assembly language programs.

Mathematical routines for trigonometric functions, exponentiation, logarithm, square root, and random numbers have been written in MAINSAIL, accurate to at least 17 decimal digits in most cases. Since they are written in MAINSAIL, there are of course no assumptions regarding word size or representation. The obscurity of their assembly language counterparts is in stark contrast to the clarity with which the algorithms are expressed in a high-level language, and has probably contributed to the astounding number of times they have been written, over and over again, for different machines. The same can be said of the MAINSAIL routines for conversion between string and floating point numbers.

MAINSAIL has a well-developed i/o capability, including any number of sequential and random files, and terminal interaction. File names are represented as strings, and the format of these strings is transparent to MAINSAIL, since they are handled only by machine-dependent routines. There are two types of sequential files: text and data. Text files are meant for legible text, for example a program or document. Whenever numeric or bits data is written to a text file, an automatic conversion is made to a string representation; similarly, such reads from a text file automatically scan for the proper string representation.

A data file contains machine-readable data in some machine-dependent format. Any mixture of numeric and bits can reside on a data file, presumably stored in a compact form identical to the internal representation within the computer. Since no conversion is necessary, input and output is efficient.

A random file is composed of fixed-length blocks of data, called file-blocks. Reads and writes supply a file-block number, and the entire file-block is involved in the transfer. A file-block is read into, or

written from, a memory area whose address is supplied to the read or write routine.

Files can be opened, closed, and deleted. Additional file-manipulation routines can be added for each site. Much of the i/o activity is handled in a machine-independent manner, so that only a few well-defined elementary procedures need be written for each machine.

CURRENT STATUS

MAINSAIL now runs on a PDP-10 with TENEX, and a PDP-11 with RT11. Development is under way for a PDP-10 with TOPS10, a PDP-11 with UNIX, and the IBM-370. Code has also been generated for an INTERDATA 7/16, VARIAN and NOVA. Many more machines were examined while developing MAINSAIL, and will be considered for implementation as sufficient resources are made available.

A number of projects across the country are interested in using MAINSAIL for the development of portable software. Among these are a robotics project, a mass spectrometry system, a program for chemical structure elucidation (now written in LISP), a computer-aided-instruction system for the teaching of logic, an automated cell classification laboratory, a machine-independent version of INTERLISP, and a display-oriented text editor.

APPENDIX F

SUBSYSTEMS AND DOCUMENTATION DIRECTORIES

Nancy Smith
 December 1974
 (updated April 1975)
 (updated Sept. 1975)
 (updated Oct. 1975)

The sources of available documentation for these programs will be abbreviated as follows:

TUG	Tenex User's Guide (1975 edition)
DUH	DEC Users Handbook
DAL	DEC Assembly Language Handbook
DML	DEC Mathematical Languages Handbook
HC	a hard-copy manual for the language
OL	on-line documentation which can be found by @DIR <DOC>programname.* . The following extensions are used on the <DOC> directory:
	.MANUAL complete usually fairly long manual
	.HELP or .HLP shorter summary, list of commands, etc.
	.SUPPLEMENT on-line supplement to hard-copy doc
	.UPDATE list of updates by date
	.SAMPLE sample program or output

See <DOC>A-LIST-OF-ALL-AVAILABLE-DOCUMENTS.INFO for complete details on these documents including where and how to order them.

Many of the major programs also have a <BULLETINS>programname.BBD file where messages about new developments, bugs, hints for using the program etc. are sent. These <BULLETINS> files can be read by any of the mail reading programs (READMAIL, RD, MSG, or BANANARD).

New programs or new versions of old programs will be put on <NEWSYS> for a trial period. The file <NEWSYS>NEW-SYSTEMS.INFO which is a message file will have a message about each program available. These new programs will not be included in the list of programs given here.

The HELP program obtained by typing @HELP gives assistance in finding the appropriate on-line documents for the various programs.

SUBSYS	DESCRIPTION	DOC
2SIDES	makes files for multi-columns and/or 2-sided listing	OL
ACCESS	gives a list of subsys's currently available to GUESTs	
ADDMSG	appends a msg to a specified file	
AID	algebraic interpretive dialog conversational lang.	HC
AIFAIL	assembly lang. - early version of FAIL from SU-AI	OL,HC
ALIAS	allows a dummy name to be set up for a program	
BAIL	SAIL debugger (on <SAIL>)	OL
BACKUP	short term file loss protection	OL
BANANARD	msg reading program (many extra features)	OL
BASIC	conversational programming lang. (DEC version)	OL,DML,TUG
BCPL	compiler writing and systems programming lang.	HC
BINCOM	binary comparison of files (now replaced by FILCOM)	DAL
BLIS10	compiler for system implementation (DEC version)	OL,HC,TUG
BLIS11	BLISS for the PDP11	
BLISS	compiler for system implementation (TENEXized)	OL,HC(DEC)
BOOTGT	loader for the PDP11 (GT40)	
BUDGET	budget management program (especially proposals)	OL
BYE	@BYE same as @BREAK (LINKS)	
CALENDAR	calendar management and reminder system	OL,TUG
CAM	the compare and merge program of SOUP see <DOC>SOUP.MANUAL	
CCL	concise command language	OL,DUH
CLEAN	a file by file directory clean-up program	OL
COPYM	reading/writing DEctapes	OL,TUG
CREF	cross-reference assembly listing	OL,DAL
CRSREF	TENEX cross-referencing program (outfile_infile(s))	
CRYPT5	En/Decrypts textfiles to provide security	OL
DCHANGE	character set conversion for "foreign" tapes see <DOC>DCHANGE.MANUAL and <DOC>DCHANG.HLP	OL
DCHECK	reads blocks of file into core & calls DDT to examine	OL
DDT	debugger (single-stepping added at IMSSS)	OL,TUG,DAL
DED	text-editor (designed for TENEX)	OL
DELOLD	deletes files by cutoff date of last access	OL
DELVER	deletes excess versions of files	TUG
DFTP	file transfers to and from the Datacomputer. (for certain special file storage needs)	OL
DIABLO	prints final copy of PUB-produced documents on DIABLO	OL
DIREXT	prints directory information for files sorted by file extension rather than file name	OL
DO	creates or appends a line to a reminder file	OL
DOM	effects the assembly and loading of a single MACRO program	OL
DONE	deletes a line from a reminder file	OL
DROP	similar to DELVER, deletes oldest and 2nd newest on *.*	
DSKACC	gives dsk allocation for all members of accounting groups	
DTACOP	DEctape to DEctape copy	
DUMPER	reads/writes magnetic tapes	
EOFIX	deletes any pages past end of file mark	OL
EXTR	"EXTRactor" processes MACRO/FAIL source files to produce .FAI listing of labels defined	
F40	FORTTRAN IV (see also <DOC>FORTTRAN.HELP and <DOC>LISP-FORTTRAN-INTERFACE.HELP)	OL,TUG,DML
FAIL	assembly language (BBN version of FAIL) (see also JSYS manual & <DOC>SUMEX-JSYS'S.INFO)	OL,HC

FED	the final edit program of SOUP see <DOC>SOUP.MANUAL	
FILCHK	checks SAIL programs for loader incompatibilities	OL
FILCOM	complete file comparison package	OL,DAL,TUG
FILDMP	dumps files in variety of formats	OL
FILES	multiple to multiple copies, renames, protections	
FILEX	for file transfers converts between DEC machine formats for dsk and DEC-tape.	OL,DUH
FORMAT	makes table of contents & index for SAIL sourcefiles	OL
FORTRA	FORTTRAN10(version 4) (see also <DOC>FORTTRAN.HELP)	OL,HC
FREQ	ranks words in text file according to frequency	
FRKCOM	compares an address space with address space of file	TUG
FTP	ARPANET file transfers	TUG
FUDGE2	updates/manipulates files containing rel programs	DAL,TUG
GETDMP	loads into core .dmp file from SU-AI (SAV only to 677777) type filename to * prompt	
GRIPE	sends comments or complaints about system to staff	TUG
HELP	helps locate on-line documentation	
HOSTAT	prints network site status information	TUG
IDDT	DDT for inferior forks	TUG,OL
IFAIL	assembly language (IMSSS version of FAIL)	OL,HC
ILISP	UC Irvine LISP (extension of LISP 1.6)	OL
IMSSS	direct link to IMSSS	
INSPEX	checks files for wasted space and pages past eof	OL
KILL	closes all jfns--useful when RESET can't get a file closed	
LAST	Gives date, time of last full dump, archive or daily dump	
LD	prints SYSTAT-like info	
LINK10	DEC loader	OL,DAL,TUG
LINK11	linker for PDP11 DOS operating system	
LINKSTAT	prints status of IMSSS link	
LISP	INTERLISP-see also <DOC>LISP-FORTRAN-INTERFACE.INFO	OL,HC
LOADER	(from IMSSS)-see <DOC>LINK10-LOADER-DIFFERENCES.HELP	TUG
LOADGT	GT40 standard format loader	
LOADVT	loader for PDP11 (GT40)	
LOWCASE	converts a text file to lowercase	
LPTSTS	gives the files on the lineprinter queue & their size	OL
MAC11	MACRO cross-compiler for the PDP11	
MACRO	assembly lang-JSYS manual & <DOC>SUMEX-JSYS'S.INFO	TUG,DAL
MAILBOX	to reroute mail (not fully implemented yet)	OL
MAILSTAT	info on queued mail	TUG
MANTIS	Fortran debugger	
MATHLAB	interactive symbolic algebraic system	OL
MLAB	mathematical modeling and graphics package	OL
MSGFIX	TECO routine to help fix the format of messages	
MTACPY	magtape program	TUG
MTCOPY	DEC magtape program	OL
MULTI	multiple-fork supervisor--switches between forks	
MY-ACCOUNTS	prints user's valid accountnames	
NDIR	gives compact list of files on connected directory	
NETSTAT	prints info on ARPANET status	TUG
NEWFILES	directory information for files written in last 24 hrs	OL
NEWINFO	gives all new files on public directories or for any file group (includes number of reads for each file)	OL
NODE	gives the geographical location of a TYMNET node	
NON	zero-compresses file, options to remove linenumbers, pagemarks, convert eol's, etc.	

PCSAMP	measures the operation of other user programs	TUG
PDP6DT	DEC-tape program	
PIP	DEC utilities program	OL,DUH
PIP11	transfers PDP11 DOS DECTapes to/from TENEX files	OL
PNTMAK	converts underlines to suitable format for LPT:	OL
POET	text editor designed for TENEX use	OL
PPL	an interactive extensible programming lang.	TUG
PROFIL	gives freq of execution of SAIL statements	OL,HC
PUB	document preparation lang.	OL
PUB2	2nd pass of PUB -- used separately to change underlines	
RD	mail reading program (MSG is better)	TUG
READMAIL	mail reading program (MSG is better)	TUG
RECOG	when ordinary recognition is ambiguous RECOG gives the possible filename matches	OL
RECORD	for pseudo-ttys, typescript of job, detaching from running job	OL
REDUCE	symbolic algebraic language	OL
RPURGE	requires confirmation before purging (delete & expunge) & puts info on purged files in a file by date	OL
RSEXEC	restricted access only	TUG
RTTY	types out a file starting at the end (reverse)	OL
RUNFIL	uses file instead of tty for input commands	TUG
RUNOFF	document-preparation language (DEC not BBN version)	OL
SAIL	ALGOL-like lang.-see also <DOC>LEAP.MANUAL	OL,HC
SCAN	scans multi-directories for a variety of file info	OL
SEARCH	searches multi-text files for English words or SAIL identifiers, can be used with TV editor	OL
SEARCHDIR	substring search of directory information on files	OL
SEARCHP	substring search also allows random reading of file	OL
SEGSAV	reads .shr & .low files to produce TENEX .sav	OL
SITBOL	compiler version of SNOBOL	OL
SNDMSG	message sender	OL,TUG
SNOBOL	string-processing programming lang.	OL,HC
SORT	stand alone COBOL column-oriented text file sorter	OL,TUG
SOS	text editor	OL
SPELL	spelling checker/corrector for text files (not TENEX)	OL
SPSS	Statistical Package for the Social Sciences	OL
SRCCOM	compares text files	TUG
STP	Western Michigan University StaTistical Package	OL
SUBMIT	submission to batch (see <DOC>BATCH.HELP)	
SWITCH	switches the format of a reminder file	OL
SYSDPY	gives SYSTAT-like info constantly updated on display (CRT) terminal	OL
SYSIN	executes LISP SYSOUT's	OL
TABLE	creates conversion tables for DCHANGE	
TALK	used with LINK command to eliminate need for ;'s	
TAPCNV	reads card image file processed by MTACPY	TUG
TBASIC	TENEXized version of DARTMOUTH BASIC	OL
TCTALK	teleconferencing over ARPANET	OL
TECO	text editor (see TENEX TECO manual)	OL,TUG
TELNET	restricted access only	TUG
TIPCOPY	sends text files to a TIP port	TUG,OL
TMERGE	merges specified text pages from files into new file	OL
TODAY	lists the contents of today's reminder file	OL
TRITAP	processes magtapes from XEROX, IMSSS, BBN	OL

TTYTRB	used to report terminal line problems	TUG
TTYTST	prints test patterns for diagnosing terminal	TUG
TV	text editor for TEC and DATAMEDIA displays	OL
TVFIX	restores bad TV files (see <DOC>TV.MANUAL)	
TYMSTAT	(for TYMNET lines only) gives measure of current efficiency of TYMNET transmission	
TYPBIN	does an octal dump of a packed file	TUG
TYPEIN	appends type-in to file with some editing allowed	OL
TYPREL	analyzes contents of .REL files	TUG
UPCASE	converts an entire file to uppercase	
WATCH	continuous on-line monitoring of system activity	TUG
WATCH.IMS	IMSSS version of WATCH	
WHAT	lists the contents of a reminder file	OL
WHO	prints SYSTAT-like information	
WHOIS	looks up username & prints name/address info on user	OL
VIEW	examines a file word by word, several typeout modes	OL
XED	text-editor (used with BANANARD)	OL
XT	reformats and prints text file	OL
Z	logs jobs off including from inferior (lower) forks & prints a witty saying	

<DOC> DIRECTORY LISTING

The following is a listing of the <DOC> directory which contains most of the on-line formal documentation about the system and subsystems.

<DOC> 13-MAY-76 08:19:25

FILE NAME	SIZE (COMPUTER PGS)	
.HELP;2	1	
2SIDES.HELP;3	3	
A-GENERAL.HELP;12	2	
A-GUIDE-TO-TENEX-USER'S-GUIDE.INFO;2		5
A-LIST-OF-AVAILABLE-DOCUMENTATION.INFO;8		14
A-SURVEY-OF-THE-DEC-HANDBOOKS.INFO;10		5
ACCOUNT-NAME-USAGE.INFO;2	3	
AID.HELP;4	2	
.INFO;3	1	
ALL-SUBSYS'S-AVAILABLE-AT-SUMEX.INFO;8		7
BACKUP.HELP;2	2	
BAIL.HELP;5	1	
.MANUAL;3	17	
.UPDATE;1	3	
BANANARD.HELP;1	1	
BANK.MANUAL;2	46	
BASIC.HLP;2	2	
.UPDATE;2	12	
BATCH.HELP;3	3	
.UPDATE;2	4	
BLIS10.HLP;4	2	
.UPDATE;2	10	
BLISS.HELP;2	2	
BSYS.MANUAL;3	25	
BUDGET.MANUAL;7	9	
.UPDATE;2	1	
.SMP;2	1	
CALENDAR.MANUAL;2	6	
CCL.HELP;2	2	
CHECKDSK.HELP;3	4	
CHESS.HELP;1	3	
CLEAN.HELP;1	2	
COPYM.HELP;2	5	
CREF.HLP;1	1	
.UPDATE;2	2	
CRYPT5.HELP;1	2	
DCHANG.HLP;2	2	
DCHANGE.MANUAL;1	12	
DCHECK.HELP;1	1	
DDT.SUPPLEMENT;1	2	
.HELP;1	1	
.BRIEF;2	4	
.SUMMARY;1	9	

DEC-HANDBOOK-GLOSSARY-UPDATE.INFO;1		3
DEC/TENEX-COMMAND-EQUIVALENTS.INFO;4		11
DED.MANUAL;1	15	
DELOLD.HELP;1	1	
DESCRIPTION-OF-SUMEX-AIM-PROJECTS.INFO;3		4
DFTP.HELP;3	5	
DIABLO.HELP;9	7	
DIREXT.HELP;1	2	
DOM.HELP;1	1	
DUMP.INFO;1	1	
EDIR.MANUAL;2	9	
.HELP;1	1	
.UPDATE;1	1	
EDIT.INFO;1	2	
EDITOR-PROGRAM-INTERFACE.INFO;2		2
EOFIX.HLP;2	1	
FAIL.MANUAL;3	70	
.HELP;5	3	
FILCHK.HELP;1	1	
FILCOM.HLP;4	1	
FILDMP.HELP;2	2	
FILEX.HLP;1	2	
.UPDATE;1	4	
FLECS.HLP;1	2	
FORDDT.HLP;1	1	
.UPDATE;1	2	
FORMAT.HELP;1	3	
FORTRA.HLP;1	1	
FORTRAN.HELP;2	11	
FTP.UPDATE;1	3	
.ANONYMOUS-ACCESS;1	3	
GLOB.HLP;1	1	
.UPDATE;1	2	
GRUMP.HELP;1	1	
GT40-LIGHTPEN.HELP;1	3	
GT40-LIGHTPEN-IMPL.DOC;1	8	
GT40-OMNI-MONITOR-DIRECTIONS.HELP;1		1
GT40-OMNIGRAPH.INFO;1	2	
GT40/OMNI-MONITOR.DOC;2	2	
GUEST-ACCESS-SUMEX.INFO;1	1	
GUEST-LOGIN.HELP;1	1	
HOW-TO-UPDATE-DOC.INFO;3	3	
IDDT.HELP;1	8	
ILISP.MANUAL;1	116	
.TENEX-MANUAL;1	49	
.HELP;2	2	
INSPEX.HLP;1	1	
INTERROGATE.HELP;4	2	
INTRO-TO-SUMEX-AIM-TENEX.INFO;5		6
ISAIL.HELP;1	1	
JSYS-INDEX.INFO;1	5	
LEAP.MANUAL;3	15	
LINK10.HLP;1	2	
.UPDATE;3	8	
LINK10-LOADER-DIFFERENCES.HELP;1		6

LISP.HELP;3	2	
.UPDATE;5	4	
LISP-FORTRAN-INTERFACE.HELP;2		2
LIST.HELP;3	6	
LOADER.UPDATE;1	2	
LPTSTS.HELP;1	2	
MACRO.HLP;1	1	
.UPDATE;3	11	
MAILBOX.HELP;1	2	
MAKLIB.HLP;1	1	
MARK-MSG.S.HELP;1	2	
MATHLAB.HELP;3	4	
MLAB.HELP;1	14	
MSG.MANUAL;4	17	
.UPDATE;3	5	
MTCOPY.HLP;2	1	
MULTI.HELP;1	1	
NEW-SOS-TO-SUMEX-SOS-COMPARISON.HELP;3		2
NEW-VERSION-SOS.INTRO;143		4
.MANUAL;1	31	
.SUPPLEMENT;144	20	
NEWFILES.HELP;2	2	
NEWINFO.HELP;1	2	
NOTE.HELP;1	2	
OLDFILES.HELP;1	1	
OMNIGRAPH-USER'S-GUIDE.INFO;1		94
OVERVIEW-OF-COMPUTER-SYSTEM.INFO;1		2
PAGESCAN.HELP;1	1	
PCAL.HELP;3	2	
PIP.HLP;3	1	
.UPDATE;2	10	
PIP11.HELP;1	2	
PLOTTER.INFO;1	2	
PNTMAK.HELP;1	1	
POET.HELP;1	3	
.MANUAL;1	13	
PROFIL.UPDATE;2	2	
PROJECTS-AND-ASSOCIATED-USERS.INFO;69		7
PSEARCH.HELP;1	2	
PUB.MANUAL;3	62	
.HELP;5	47	
.UPDATE;10	8	
RADIX.HELP;1	1	
RECOG.HELP;1	2	
RECORD.MANUAL;3	19	
REDUCE.MANUAL;1	44	
RPURGE.HELP;1	2	
RTTY.HELP;1	1	
RUNOFF.HLP;1	2	
.UPDATE;1	24	
.COMMANDS;1	3	
.HELP;1	1	
SAIL.HELP;2	1	
.SUPPLEMENT;4	34	
.TENEX-SUPPLEMENT;2	7	

.BEGIN-MANUAL;2	57	
SAMPLE.PUB;1	2	
SCAN.HELP;1	1	
SCROLL.MANUAL;4	7	
.HELP;2	2	
SEARCH.MANUAL;3	6	
.INFO;6	2	
SEARCHDIR.HELP;1	1	
SEARCHP.HELP;1	2	
SEGSAV.HELP;1	1	
SETTING-UP-NEW-USER-DIRECTORIES.INFO;1		1
SIMCOM.HLP;1	2	
SIMDDT.HLP;1	5	
SIMRTS.HLP;1	2	
SIMULA.HLP;3	2	
SITBOL.HELP;1	18	
SNDMSG.HELP;7	3	
SNOBOL.MANUAL;1	40	
SORT.HLP;1	2	
.INFO;1	1	
SOS.HELP;5	7	
.MANUAL;1	54	
SOUP.HLP;1	2	
.MANUAL;2	13	
SPELL.MANUAL;10	7	
SPSS.MANUAL;1	11	
SRCCOM.UPDATE;1	1	
STP.MANUAL;1	83	
.INDEX;1	2	
SUMEX-JSYS'S.INFO;2	12	
SUPXEC.MANUAL;2	9	
SYSDPY.HELP;1	2	
SYSIN.HELP;1	1	
SYSTEM-SCHEDULE.INFO;4	2	
TAPE.INFO;1	1	
TCTALK.MANUAL;2	28	
.SAMPLE;1	2	
.HELP;1	4	
TECO.SAMPLE;1	1	
.COMMANDS;1	3	
.HELP;1	1	
.TEXT1;1	7	
.TEXT2;1	3	
.SUMMARY;1	4	
TEKTRONIX.HELP;1	1	
TELNET.INFO;1	2	
TENEX-EXEC-MANUAL-UPDATE.INFO;6		16
TERMINAL-LINKING.INFO;1	1	
TIPCOPY.HELP;1	1	
TMERGE.HELP;4	1	
TRITAP.HELP;1	1	
TV.UPDATE;12	9	
.MANUAL;9	26	
TYMNET-INSTRUCTIONS.INFO;4	3	
TYPEIN.HELP;2	2	

UPCASE.HELP;1	1	
USER-NAME-ADDRESS-PHONE.INFO;109		20
VIEW.HELP;2	2	
WHOIS.HELP;1	1	
XED.MANUAL;301	19	
.301-CHANGES;301	5	
.NEWS;301	1	
XSEARCH.ALGORITHM;1	8	
.HELP;2	9	
XT .HELP;1	1	

220 FILES, 1779 PAGES

<BULLETINS> DIRECTORY LISTING

The following is a listing of the <BULLETINS> directory which is a repository of informal or transient information about the system, subsystems, current events, and items of interest.

<BULLETINS> 13-MAY-76 08:20:41

FILE NAME	SIZE (COMPUTER PGS)
12-MAR-75.SYSLTR;1	3
AIM-WORKSHOP-ATTENDANCE.BBD;1	5
ARITHMETIC.BBD;1	1
ARPANET.BBD;1	2
BANK.BBD;1	1
BASIC.BBD;1	4
BATCH.BBD;1	1
BULLETINS.BBD;1	2
CALENDAR.BBD;4	1
COMPATIBILITY.BBD;1	1
CONSTANTS(PHYSICAL-OR-CHEMICAL).BBD;1	1
CRYPTION.BBD;1	2
DATA-MEDIA.BBD;1	1
DATACOMPUTER-SCHEDULE.BBD;1	1
DDT.BBD;1	1
DFTP.BBD;1	1
DO .BBD;1	1
EDIT.BBD;1	2
EMPLOYMENT-WANTED.BBD;1	1
EXCESS-BAGGAGE.BBD;1	1
FAIL.BBD;1	1
FILE-MANAGEMENT.BBD;1	3
FILES.BBD;1	6
FORTRAN.BBD;1	19
FTP.CHANGES;1	1
GOOD-LISP-USAGE.BBD;2	6
GOOD-SYSTEM-USAGE.BBD;1	2
GUEST-LIST.BBD;2	1
ILISP.BBD;1	1
IMP-PM-SCHEDULE.INFO;1	1
IN-WATS.BBD;1	1
JSYS.BBD;1	1
KWIC.BBD;1	3
LEAP.BBD;1	1
LIBRARY-SAIL.BBD;1	5
LINK10.BBD;1	1
LINKING.BBD;1	1
LISP.BBD;2	2
LIST.BBD;1	2
LOGIN-CMD.BBD;1	1
LOGIN-MESSAGES.BBD;2	58
MACRO.BBD;1	1

MEETINGS.BBD; 1	1	
MLAB.BBD; 1	3	
NEW-EXEC.INFO; 9	9	
PDP11-GT40.BBD; 1	2	
PLOT.BBD; 1	3	
PLOTTER.BBD; 1	1	
PROTECTION.BBD; 1	1	
PUB.BBD; 1	7	
RECORD.BBD; 1	2	
SAIL.BBD; 1	4	
SEARCH.BBD; 1	2	
SNDMSG-READMAIL.BBD; 1	2	
SORT.BBD; 1	1	
SOS.BBD; 1	1	
SPELL.BBD; 1	1	
TECO.BBD; 1	2	
TESTIMONIALS.BBD; 1	1	
TIMING.BBD; 1	3	
TYMNET.BBD; 1	1	
.NODE-DOWNTIME; 1	1	
TYMNET-RESPONSE-STATISTICS.JUN/75-APR/76; 1		16
USER-INTERFACE-PROTOCOLS.BBD; 1		2
WORKSHOP-DEMO-SCHEDULE.INFO; 1		1

75 FILES, 293 PAGES

APPENDIX G

SUMEX-AIM SUMMARY FOR ARPANET RESOURCES HANDBOOK

The following material was assembled as a description of the SUMEX-AIM resource for the ARPANET RESOURCE HANDBOOK (NIC 23200), Edited by E. J. Feinler of the Network Information Center at Stanford Research Institute.

(SUMEX-AIM) Stanford University Medical EXperimental Computer

(FUNCTION)

SERVER COMPUTER: PDP-10 HOST ADDR. 56 IMP 56/HOST 0

The SUMEX (Stanford University Medical EXperimental Computer) project is a cooperative computer resource involving participation by the Biotechnology Resources Branch of the National Institutes of Health and a variety of major research projects, many of which are also supported by ARPA and which thereby are authorized for access to the ARPANET.

SUMEX encompasses a dual mission: 1) supporting the development of artificial intelligence (AI) computer science research with special emphasis on biological and medical problems and 2) the demonstration of computer resource sharing within a national community. The SUMEX resource resides administratively within the Genetics Department of the Stanford University Medical School and serves as a nucleus for a growing community of projects, both within and external to Stanford. SUMEX provides computing facilities specifically tuned to the needs of AI research and communication tools to facilitate inter- and intra-group contacts as well as trial dissemination of research products to medical users. The project also develops tools for and takes an active role in stimulating community relationships among collaborating projects and medical researchers. Currently active projects span a broad range of application areas such as clinical diagnostic decision-making, molecular structure-interpretation, belief systems modeling, mental function modeling, and instrument data interpretation.

(ADDRESS)

SUMEX-AIM Computer Project, TB105
Stanford University Medical Center
Stanford, California 94305

(PERSONNEL)

(PRINCIPAL-INVESTIGATOR)

Joshua Lederberg (LEDERBERG@SUMEX) (415) 497-5801

(LIAISON)
Richard Cower (COWER@SUMEX) (415) 497-5208

(SOFTWARE-CONTACT)
TENEX system:
Andrew Sweer (SWEER@SUMEX) (415) 497-6707
Subsystems:
Nancy Smith (NSMITH@SUMEX) (415) 497-6982

(HARDWARE-CONTACT)
Richard Cower (COWER@SUMEX) (415) 497-5208

(CONSULTANT)
Richard Kahler (KAHLER@SUMEX) (415) 497-5336

(SYSTEM-USE)

(POTENTIAL USERS)

For information and introductory literature contact:

Dr. Elliott Levinthal
AIM User Liaison
SUMEX Computer Project
c/o Department of Genetics, S047
Stanford University Medical Center
Stanford, California 94305

User projects are separately funded and autonomous in their management and are selected for access to SUMEX on the basis of their scientific merit as well as their commitment to the community goals of SUMEX. Procedures for access to SUMEX are governed by a national advisory group. GUEST access is provided only for limited demonstrations of applications programs developed by the various SUMEX projects. Applications for GUEST access should be made either to Dr. Levinthal (address above) or the Principal Investigator of the particular project offering the program.

SUMEX-AIM does not sell computer time.

Long-term online storage is not available to network users.

(SERVICE-SCHEDULE)

SYSTEM DOWNTIME:

THUR -- 18:00 to 24:00 for preventive maintenance
SUN -- 6:00 to 10:00 for system backup

TYPICAL PRIME TIME LOAD = 22 users
MAX. NO. USERS = 50 users
NO. NETWORK SLOTS = 24

(LOGIN)

TELNET INFO:

- . Appropriate transmission mode = Character-at-a-time
- . Appropriate echo mode = Full-duplex; however, TENEX will also operate in half-duplex.
- . Mapping between NVT and local character set uses the full ASCII character set; <CR-LF> received from an NVT is passed to TENEX as New Line (Octal 37).
- . TENEX EXEC does not distinguish between upper and lower case alphabets and will accept either. At SUMEX the defaults are "no raise" and "lowercase". If the "raise" command is given then lower case characters received will be translated to upper case, and echoed in upper case. "No lowercase" causes lower case characters to be sent to the terminal as upper case.
- . The user can declare his/her terminal type to EXEC as follows:

```
[@]terALT[minal (type is)] TYPE <CR>
```
- . TIP settings - t e 0, e r

USER INFO:

Free experimental use is not available.

- . USERNAME = user's last name
- . ACCOUNT NAME = an assigned string
- . PASSWORD = user's assigned password

LOGIN:

Full Duplex Login (default condition)

```
Connect to SUMEX-AIM, then type:
[@]login <SP> USERNAME <SP> PASSWORD <SP> ACCT-NAME <CR>
[job xx on tty xx date time]
[previous login: date time]
[other active jobs for this user if any]
[current systems messages if any]
[next scheduled downtime]
[execution of commands from <username>login.cmd if any]
[notification of the existence of new mail if any]
[@]
```

Half-Duplex Login

```

Connect to SUMEX-AIM, then type:
[@]half <CR>
[@]login <SP> USERNAME <SP> PASSWORD <SP> ACCT-NAME <CR>
[job xx on tty xx date time]
[previous login: date time]
[other active jobs for this user if any]
[current systems messages if any]
[next scheduled downtime]
[execution of commands from <username>login.cmd if any]
[notification of the existence of new mail if any]
[@]

```

Guest Login

```

*GUESTs are users authorized to run the various applications
*programs. They are provided with a restricted version of
*the EXEC and restricted access to other resources not
*directly related to running the programs. SNDMSG capability
*is available.

```

```

Connect to SUMEX-AIM, then type:
[@]guest <SP> LASTNAME <SP> GUEST-PASSWORD <CR>
[ Checking guest registry ...]
[We would like to get some general information from you as
our guest.]

[(CTRL-X to redo the current prompt.)]
[Full name? (end with CR)]
NAME
[Affiliation? (end with CR)]
AFFILIATION
[Mailing address, phone number? (end with ^Z)]
ADDRESS
PHONE-NO.
CONTROL-Z
[From whom did you get the password? (end with CR)]
NAME
[Thank you. If you come in as a guest again, and use the]
[name "NAME", you will skip these questions.]
[Type "ACCESS" to see what programs are currently available
to guests.]

[job xx on tty xx date time]
[previous login: date time]
[current system messages if any]
[next scheduled downtime]
[@]
[@; ***** WELCOME TO SUMEX *****]
[@; Please type HELP after the @sign if you wish further
information.]
[@; NOTE: Say "RUN SNDMSG" to send messages as a guest.]
[@]

```

SUBSYSTEM INTERRUPT = CONTROL C

SUBSYSTEM CONTINUE = [**@**]con <CR> or [**@**]c <CR>

(LOGOUT)

LOGOUT:

```
[@]logout <CR>
[number of other active jobs for this user if any]
[killed job xx, user xyz, acct stuv, tty xx,
 at DATE TIME used 0:0:0 in 0:0:0]
```

After logout, a Network user should instruct his/her NCP to close both connections.

AUTOLOGOUT:

Breaking Network connections does not log the user out; however, his/her job becomes "detached". If after 20 min. the job has not been reattached (via the "attach" command), the job is logged out.

(CONTROL-CHARACTERS)

A few ASCII control characters are listed below:

Delete last character	CONTROL-A (or DEL key)
Delete previous word	CONTROL-W
Delete command or line	CONTROL-X
Abort print	CONTROL-C
Retype edited command	CONTROL-R
Prompt or help	?
Force recognition	ALTMODE
Is-system-still-there?	CONTROL-T

For a complete set of control characters available in TENEX see BBN TENEX EXEC Language Manual. However, note that the DELETE (RUBOUT) key at SUMEX is used for deletion of a single character not abortion of the entire typein as in standard TENEX.

(HELP)

All TENEX commands available to the user are documented online. The user may obtain a complete list of these commands by typing:

```
[@]?
```

At any point TENEX appears to expect a word or argument, the user may type "?" and a list of allowable keywords or arguments will be displayed.

Also, at SUMEX, help with using the system or the various programs may be obtained by typing:

```
[@]help <CR>
```

(NETWORK-COMMANDS)

(LIST-ACTIVE-USERS)

Connect to SUMEX-AIM, then type:
 [@]systat <CR>

(NETWORK-STATUS)

GENERAL STATUS

Connect and Login to SUMEX-AIM, then type:
 [@]netstat <CR>
 [*] <CR>

NETWORK TENEX LOAD STATUS

Connect to SUMEX-AIM, then type:
 [@]netload <CR>

(LINK-TO-ACTIVE-USERS)

To link to an active user on a given TENEX host,
 connect to that host, then type:
 [@]link <SP> ACTIVE USERID <CR>
 ACTIVE TTY-NO.<CR>(number - not word 'tty')
 [@];...MESSAGE...<CR>
 (NOTE: each line must start with a ';' and end with <CR>)
 [@];...REPLY...<CR> (Text of reply)
 [@]break <CR> (breaks link and returns user to EXEC)

To answer a link from an active user, type:
 [link from smith, job x, tty nn]
 [@;this is smith, how are you]
 CONTROL-C (ONLY if not already in EXEC)
 [@];...REPLY...<CR>

To refuse all links, type:
 [@]refuse <CR>

Users from other TENEX sites can link to users at
 SUMEX through the RSEXEC. Users at SUMEX require special
 authorization for access to RSEXEC to initiate links.

(SEND-MESSAGE)

Login to TENEX, then type:
 [@]sndmsg <CR>
 [to:] USERNAME@HOSTNAME <CR>
 (NOTE: Here the user must actually type an "@"
 followed by the HOSTNAME of the recipient.
 If coming through a TIP, type two '@' signs.)
 [cc:] USERNAME@HOSTNAME,USERNAME@HOSTNAME,etc. <CR>
 [subject:]...HEADER or TITLE.....<CR>
 [message:].....TEXT.....<CR>
 .(edit with control-A,Q,R,X or DEL).<CR>
 ... (call TECO to edit or.....<CR>
 ...insert file with CONTROL-B)....<CR>
end with CONTROL-Z
 [q,s,?,carriage-return:] <CR> sends the mail at once

Q delivers mail later
? lists other options

(GRIPE)

Login to SUMEX-AIM, then type:
[@]gripe <CR>
[gripping on subject of] general <CR>
[message (? for help):]
.....MESSAGE.....
CONTROL-Z
[thank you for your comments]
[@]

(RETRIEVE-MESSAGE)

During TENEX login this statement will appear:
[you have a message]
Also, at SUMEX, at regular intervals, the message file
will be checked and this statement will appear:
[you have new mail]

To retrieve the message type:
[@]readmail <CR> <CR>
[;<FILENAME>message.txt;1 DATE TIME
SENDER]
[.....TEXT.....]

For interactive reading and deletion of selected
messages use the BANANARD program rather than READMAIL
[@]BANANARD <CR>
-< (help is available on-line by typing ?).

To delete all messages from your directory, type:
[@]delete message.txt <CR>
[@]expunge <CR>

(CONSULTATION)

TENEX offers two ways to send messages to system
programmers. They are the subsystems GRIPE and SNDMSG, and
are obtained by typing the appropriate subsystem name to
the EXEC. Each subsystem provides self-explanatory
instructions. GRIPE is generally used for constructive
criticism about a subsystem or TENEX. In general gripes
are low level criticisms to which formal responses are not
generally made. SNDMSG should be used for direct questions
to specific individuals. If a user does not know which
specific individual to contact, a message can be sent
via SNDMSG to SUMEX@SUMEX. The message file on this
directory will be read by the consultant (to be appointed
shortly) and redirected to the appropriate member of the
systems staff for action.

(COMMAND-SUMMARY)

To obtain a complete list of commands that exist in the TENEX EXEC login to SUMEX, then type:

[@] ?

TENEX commands are fully documented in the TENEX EXEC Manual (Ref. 4).

General help is also available through the HELP program:

[@]help <CR>

(FILE-NAMING)

File names in TENEX are composed of five identifiers. These are device, directory name, file name, extension, and version. These five items uniquely identify any file accessible to a user on the system. The device name identifies which device in the system contains a given file. The directory name gives the directory under which the file appears. The file name, extension, and version identify a particular file in a given directory. Here is an example of a TENEX file name:

DSK:<SUBSYS>TECO.SAV;1

The HELP program contains pointers to general information files available on-line as well as pointers to documentation files for specific programs. It also gives information on the assignment of filenames to public files at SUMEX so that they can be easily located.

(PROTOCOLS)

(SERVER)

Network Server Protocols currently implemented are:

1. TELNET (Network Standard) (ICP to Socket 1 for old protocol TELNET or Socket 27 for new protocol TELNET) Establishes a NVT connection to TENEX.
2. FTP (Network Standard) (ICP to Socket 3). Establishes a duplex connection to the File Transfer Protocol Server. SUMEX supports anonymous access to a selected set of directories--login with username as ANONYMOUS and use lastname as the password.
3. TENEX-TENEX (Private Standard) (Socket 105 octal) Used for ICP to file transfer service.
4. MAXIM (Private Standard) (ICP to Socket 21). Transmits a TENEX "maxim".
5. RSSER (Network Standard) (ICP to Socket 365 octal) establishes a duplex connection to RSEXEC server process.
6. RSEXEC (Network Standard) (ICP to

Socket 367 octal). User is connected to the RSEXEC which may be used to access network news, host status, etc.

7. DAYTIME (Private Standard) (ICP to Socket 15)
Transmits day and time in full format.

(USER)

1. FTP (Network Standard)
To access, type:
[@]ftp <CR>
[*]help <CR>

(NCP-INTERFACE-FROM-LOCAL-PROGRAMS)

The NCP is implemented within the TENEX file system, and hence a Network connection appears to the assembly-language programmer as a sequential file whose byte size is that of the connection. The usual file JSYSes - openf, closf, bin, bout, etc. - are used to manipulate the connection.

Network connections are distinguished from other TENEX files by their file names, in which local socket number, and remote host and socket number are embedded. See the TENEX JSYS Manual for further information.

(RESOURCES)

(HARDWARE)

(COMPUTER)

TYPE	CORE AMOUNT	CORE SPEED	WORD LENGTH
(2)PDP-10 KI	256K	1 microsec	36 bits

(PERIPHERALS)

	HOW MANY	TYPE	MAKE	MODEL
(DISKS)	6	DOUBLE DENSITY	DEC	RP03
(DRUMS)	2	FIXED HEAD	DIG. DEVEL. CORP.	A7312-D-8
(TAPES)	2	9 TRACK	DEC	TU30
	2	DECTAPE	DEC	TU56
(PRINTER)	1	ROTATING DRUM	DATA PRODUCTS	2410
(PLOTTER)	1	100 STEPS/INCH	CALCOMP	565

(TERMINALS)

SUMEX supports a wide variety of terminals. The most commonly used display terminal is the DataMedia for which a specially designed keyboard is available to interface with the TV-editor used at SUMEX. A variety of other software programs are being developed with special display features designed for use on the DataMedia terminal.

(OPERATING-SYSTEM)

TENEX is a virtual-memory operating system for a time-shared DEC PDP-10 computer that provides a 256K word virtual address space to each process. It permits the creation and running of hierarchies of interdependent processes, accommodates large numbers of users, has extensive file system capabilities, and a well human-engineered executive command language. Most programs written for the standard DEC monitor (10/50 code) run directly.

SUMEX runs a special version of TENEX, modified for the KI-10 computer from the original BBN KA-10 version, to accommodate the KI-10 paging hardware. Preliminary modifications to TENEX version 130 for the KI-10 were made by Dan Murphy at DEC. Under Rainer Schulz, that system was extensively debugged and updated to version 131 (at the NASA Ames Research Center) as well as substantially improved in throughput (at Stanford: Institute for Mathematical Studies in the Social Sciences and SUMEX). This version of KI-TENEX currently operating at SUMEX has approximately twice the throughput of KA-TENEX systems. The staff is currently debugging a dual processor version of TENEX.

SUMEX has a broad range of users including many computer novices. To facilitate smooth interface with our community of users, we have made a number of local modifications to TENEX, particularly the EXEC. Many of these are in the area of user-settable options. A complete list and description of these modifications to the standard TENEX EXEC is available online in <DOC>TENEX-EXEC-MANUAL-UPDATE.INFO .

(USER-PROGRAMS)

(APPLICATIONS PROGRAMS)

These offerings are expected to increase as our newer projects become established. A list of the current programs is available to GUESTS by typing "ACCESS".

DENDRAL PROGRAMS:

CONGEN

(TYPE)

Chemistry: Computer-Assisted Structure Elucidation

(CONTACT)

JOHNSON@SUMEX or SMITH@SUMEX

(DESCRIPTION)

Congen (CONstrained structure GENeration) accepts as input known structural features of an unknown molecule (whose

elemental composition is known) and produces all structural isomers consistent with these data. The features and constraints are entered in an interactive session with the program and results can be drawn at a terminal or further constraints added based on examination of new data. CONGEN represents an initial version of a program for computer-assisted structure elucidation. The structure generator which underlies CONGEN has been described (see Masinter et. al., J. Amer. Chem. Soc., 96, 7702 (1973) and *ibid.*, 7714 (1974)).

(ACCESS)

For GUESTs: [@]congen <CR>
For Regular Users: [@]sysin <dendral>congen <CR>

(DOCUMENTATION)

<DENDRAL>CONGEN.DOC

INTSUM

(TYPE)

Chemistry: Mass Spectrometry

(CONTACT)

JOHNSON@SUMEX or SMITH@SUMEX

(DESCRIPTION)

Given a set of known, related structures and the mass spectrum corresponding to each structure, INTSUM suggests possible fragmentation processes which resulted in the observed ions, and then summarizes the results in terms of processes which are general to the class of structures, and those which are specific to certain members of the class. (See Smith et. al., Tetrahedron, 29, 3117 (1973)).

(ACCESS)

For GUESTs: [@]intsum <CR>
For regular users: [@]sysin <dendral>intsum <CR>

(DOCUMENTATION)

<DENDRAL>INTSUM.DOC

MOLION

(TYPE)

Chemistry: Mass Spectrometry

(CONTACT)

JOHNSON@SUMEX or SMITH@SUMEX

(DESCRIPTION)

Molecular ion determination program. Given a (low or high resolution) mass spectrum in which the molecular ion may or may not be present, this program suggests a ranked list of candidate molecular ions. (See G. Dromey, B. G. Buchanan, D. H. Smith, J. Lederberg and C. Djerassi, J. Org. Chem., 40, 770(1975)).

(ACCESS)

For GUESTs: [@]molion <CR>
For Regular Users: [@]sysin <dendral>molion <CR>

(DOCUMENTATION)

<DENDRAL>MOLION.DOC

PLANNER

(TYPE)

Chemistry: Mass Spectrometry

(CONTACT)

JOHNSON@SUMEX or SMITH@SUMEX

(DESCRIPTION)

Planner infers possible structures of unknown compounds (singly or as mixtures) given a mass spectrum and fragmentation rules of the class of compounds to which the unknown(s) presumably belongs. (See Smith et. al., J. Amer. Chem. Soc., 94, 5962 (1972); ibid., 95, 6078 (1973)).

(ACCESS)

For GUESTs: [@]plan <CR>
For Regular Users: [@]sysin <dendral>plan <CR>

(DOCUMENTATION)

<DENDRAL>PLAN.DOC

MYCIN PROGRAMS:

MYCIN

(TYPE)

Interactive Consultation for Infectious Disease Patients

(CONTACT)

SCOTTE@SUMEX

(DESCRIPTION)

MYCIN is an interactive program which utilizes data available from the microbiology and clinical chemistry laboratories, plus the physician's response to computer-generated questions, to provide physician nonspecialists with consultative advice on diagnosis and antimicrobial therapy for patients with bacterial infections. The program also has interactive capabilities allowing the user to request explanations of the consultation program's actions and reasoning, to ask about MYCIN's static knowledge base or about specific conclusions made during the consultation, and to teach the program by entering new pieces of judgmental knowledge.

(ACCESS)

For GUESTs: [@]mycin <CR>
 For Regular Users: [@]<mycin>mycin <CR>

(DOCUMENTATION)

<MYCIN>MYCIN.DOC

HIGHER MENTAL FUNCTIONS MODELING PROGRAMS:

PARRY

(TYPE)

Interactive simulation of paranoid thought processes

(CONTACT)

COLBY@SUMEX or FAUGHT@SUMEX or PARKISON@SUMEX

(DESCRIPTION)

PARRY is an interactive simulation of a model of paranoid thought processes. The purpose of the user (usually a

clinical psychiatrist) is to conduct a first interview with the patient (PARRY) and obtain a diagnosis. The interview usually lasts 40-60 input/output pairs. The program is divided into two modules: the recognizer module and the response module. The recognizer accepts natural language input expressions (English) to determine its semantic content. The response module uses the model's inference, affect, and intentions mechanisms to determine the appropriate response.

(ACCESS)

For GUESTs: [*@*]parry <CR>
For Regular Users: [*@*]<parry>parry <CR>

(DOCUMENTATION)

<PARRY>PARRY.DOC

GENERAL UTILITIES PROGRAMS FROM NIH:

MLAB

(TYPE)

Mathematical Modeling

(CONTACT)

JOHNSON@SUMEX

(DESCRIPTION)

MLAB stands for MODELAB which stands for "modeling laboratory". It is a program which allows one to do scalar and matrix computations, curve-fitting, and differential equation solving. Graphic facilities are also provided. This program was written by, and obtained from, Gary Knott at DCRT/NIH.

(ACCESS)

For GUESTs: [*@*]mlab <CR>
For Regular Users: [*@*]mlab <CR>

(DOCUMENTATION)

<SYSSUP>MLABD.TXT (also available as interactive help)

OMNIGRAPH

(TYPE)

Terminal Independent Graphics Software

(CONTACT)

JOHNSON@SUMEX

(DESCRIPTION)

Omnigraph is a graphics subroutine package, designed by Robert F. Sproull while at DCRT/NIH, for driving a number of different display devices with either SAIL or FORTRAN as the programming language. The Omnigraph system is designed for routine graphics applications, not for high-performance terminals. Terminals which can be used with this package include Dec GT/40, Dec 340, Tektronix 4010, Computek, and the Ards.

(ACCESS)

Used by linking appropriate language dependent code in with the program which incorporates the Omnigraph calls. Terminal type is defined at run-time, and appropriate code is loaded by the system. Language dependent code segments are available as LIB:DISSAI for SAIL, LIB:DISFOR for F40 Fortran, and as LIB:DISF10 for FORTRAN-10.

(DOCUMENTATION)

<DOC>OMNIGRAPH-USER'S-GUIDE.INFO

RUTGERS RESEARCH RESOURCE ON COMPUTERS IN BIOMEDICINE:
[To be supplied later]

(LOCAL SOFTWARE PROGRAMS)

SUMEX is willing to provide copies of any of its non-contract software programs to other interested sites. The following is a list of those programs which originated at IMSSS or SUMEX with SUMEX as their network distribution point that should be of particular interest to other TENEX sites. To facilitate the most efficient distribution of these programs, we request that any site desiring one of the programs appoint a single person to serve as the local contact with SUMEX for that program. The designated person should communicate directly with the author of the program (listed below). In this way, we can insure that the interested site receives a correct working version of the program as well as any updates as they become available. We also request that any subsequent comments or bug reports be channeled to SUMEX through the local contact. If local modifications need to

be made, we will be happy to incorporate them under conditional compilation switches in the master sources thereby simplifying the procedure of moving to updated versions in the future. We cannot be responsible for maintaining versions of these programs obtained in any other manner.

TENEX SAIL:

SAIL which was developed at SU-AI is an ALGOL-like language with complex data- and control-structure extensions for artificial intelligence research and it compiles reasonably efficient code. Robert Smith at IMSSS has TENEXized SAIL. It has been given complete access to the facilities provided by TENEX and a number of new features (random input-output, interrupt system) have been added. These changes were merged into the master-files at SU-AI insuring integrity of the SAIL software.

SUMEX and IMSSS are currently organizing a SAIL library. Anyone interested in contributing routines should contact DANIELS@SUMEX.

A number of utilities to support SAIL usage are also available at SUMEX including:

BAIL written by John Reiser at SU-AI which is an interactive runtime debugging package

PROFIL written by D. Sweet which gives the frequency of execution of SAIL statements

FILCHK written by R. Smith which checks SAIL programs for loader incompatibilities

FORMAT written by R. Smith which adds a table of contents and optional index to SAIL programs

Contact RSMITH@SUMEX for details.

TENEX UCI-LISP:

UCI-LISP is an extension of LISP 1.6 with many new features. It has been TENEXized at IMSSS by Tom Wolpert. The pmapped IO is extremely fast. It also includes the edit and break packages of 1972 INTERLISP. TENEX UCI-LISP is approximately 6 times faster than INTERLISP. Complete documentation is available. Contact WOLPERT@SUMEX for details.

MACHINE-INDEPENDENT SAIL

----- ----

A machine-independent compiling system is being developed for a major subset of the SAIL language by Clark Wilcox at SUMEX. Compilers have been created for a number of machines, including the PDP-11, PDP-10, IBM-SYSTEM/360, and NOVA. The compiler and much of the runtime system are written in SAIL, so that the system is portable. This system is still developmental but will be available in the near future. Inquiries should be addressed to WILCOX@SUMEX-AIM .

RECORD:

The RECORD program by R. Smith creates pseudo-teletype jobs. It requires the pseudo-teletype code developed at AMES which is not supported in the standard BBN TENEX system. It can be used for running 3 jobs simultaneously from the same terminal with easy switching back and forth. It optionally makes a typescript of the entire session on the pseudo-teletype which is very nice for preparing documentation on program use, keeping a record of applications users, recording the action of program bugs, etc. RECORD also allows for detaching from running jobs with a very large buffer for storing program output until the job is reattached. Contact RSMITH@SUMEX for details.

TV:

TV is a display-oriented editor for use on DATAMEDIA, TEC, IMLAC and several other display terminals. It was written by Pentti Kanerva at IMSSS and is a relative of several other TV-editors developed at IMSSS and SU-AI. It provides many features for updating text files, such as good cursor control user-defined macros, and string searching. The latest version of TENEX SAIL is required to compile the editor. Contact P. Kanerva through NSMITH@SUMEX for specifics on which terminals are or could be supported by the additional of a terminal dependent front end for the editor. An older version of the editor is also available for non-TENEX PDP10 sites.

PUB Macros Package and Documentation:

--- -----

PUB is a document preparation language written by Larry Tesler formerly of SU-AI and currently at XEROX-PARC. A complete set of PUB macros for easy use of PUB has been written by Nancy Smith at SUMEX with a full manual describing the use of this package. The macros are designed to produce interestingly formatted documents by relatively inexperienced

computer users. It is designed for a non-XGP TENEX site.
Contact NSMITH@SUMEX for details.

DDT:

Robert Smith at IMSSS has added single-stepping to TENEX DDT.
Contact RSMITH@SUMEX for details.

(NEW DEC RELEASES WITH TENEX MODIFICATIONS)

SUMEX has purchased the latest FORTRAN10 package from DEC and modified both the programs and PA1050 to get it running on TENEX. SUMEX, of course, cannot share the sources but would be happy to share the modifications with any other TENEX sites who are DEC-authorized users of the new FORTRAN10. The following is a partial list of the late release DEC software running at SUMEX:

FORTTRAN10	Version 4
F40	Version 27
LINK10	Version 2A
MACRO	Version 50
BLIS10	Version 5
RUNOFF	Version 10

(OTHER LICENSED SOFTWARE)

SITBOL (Stevens Institute of Technology version of SNOBOL)

(STANDARD SOFTWARE PROGRAMS)

The following are the major standard TENEX and/or DEC programs routinely offered at PDP10 sites which are available at SUMEX plus an assortment of programs which we have obtained from other network sites. In some cases, local modifications have been made. The documentation for these programs can be located with the assistance of the HELP program.

ADDMSG	DCHANGE	FRKCOM	MACRO	REDUCE	SYSIN
AID	DDT	FTP	MAILSTAT	RUNFIL	TALK
BANANARD	DED	FUDGE2	MAILSYS	RUNOFF	TAPCNV
BASIC	DELVER	GRIPE	MULTI	SAIL	TCTALK
BCPL	DUMPER	HOSTAT	NETSTAT	SDDT	TECO
BINCOM	F40	IDDT	PA1050	SNDMSG	TTYTRB
BLIS10	FAIL	ILISP	PCSAMP	SNOBOL	TTYTST
BSYS	FED	INTERLISP	PIP	SORT	TYPBIN
CALENDAR	FILCOM	LD	POET	SOS	TYPREL
CAM	FILDMP	LINK10	PPL	SPELL	UDDT
CCL	FILES	LOADER	PUB	SRCCOM	WATCH
COPYM	FILEX	LOADGT	RD	SUBMIT	XED

CREF FORTAN10 LOADVT READMAIL
 (LOCAL UTILITIES PROGRAMS)

SUMEX also has a number of local utilities programs which we would be happy to share with other sites but for which we are unable to provide maintenance or guaranteed further support. (Many of these programs are from IMSSS).

2SIDES	makes files for multi-column & 2-sided listing
ACCESS	gives a list of programs available to guests
BACKUP	short term file loss protection
BUDGET	budget preparation program
CARDS	creates online "card catalog" for a library
CLEAN	file by file directory clean-up
CRYPT5	en/decrypts text files
DCHECK	reads blocks of file into core & examines with DDT
DIREXT	lists directory ordered by file extension
DIRNUM	translates directory name to no. for DEC programs
DO	creates or appends a line to a reminder file
DONE	deletes a line from a reminder file
FREQ	ranks words in text file according to frequency
LOWCASE	converts a text file to lowercase
NEWFILES	prints info on all user's files written in 24 hrs.
NEWINFO	prints info on all public files written in 24 hrs.
NON	zero-compresses, removes linenos., pagemarks, etc.
PERUSE	allows fast reading of random parts of a file
RPURGE	interactive selection of files to purge--records write/create/purge date--optional comments on file
RTTY	types out a file starting at the end (reverse)
SEARCHDIR	substring search of directory information--like wildcard for names--also searches on author & date
SEARCHP	substring search of multi-files with PERUSEing of random parts of the file
TBASIC	TENEXized version of Dartmouth Basic
TMERGE	merges specified pages from file(s) into new file
TODAY	lists the contents of today's reminder file
UPCASE	converts a text file to uppercase
WHAT	lists the contents of a reminder file
WHOIS	looks up username & prints name/address info
XSEARCH	very fast substring search of multiple files with optional production of "hit" list for TV-editor
Z	logout from lower forks.

(INTERESTS)

(SUMEX STAFF)

The interests of the SUMEX staff center around two themes:
 1) providing easy access to the system for a community of remote users with widely differing computer experience including medical professionals with no previous computer experience and
 2) developing means to facilitate communication and resource sharing among the various projects. Therefore, areas of interest and program development include: message sending and

reading facilities, creation of an on-line bulletin board, organization of on-line documentation with a help system for easy access of the material, development of libraries of program routines, acquiring of available utilities packages, and study of techniques for acquiring and employing user models in all of these areas.

(SUMEX PROJECTS AND THEIR PRINCIPAL INVESTIGATOR(S))

(STANFORD)

DENDRAL

Prof. C. Djerassi (Chemistry)
 Prof. J. Lederberg (Genetics)
 Prof. E. Feigenbaum (Computer Science)

MYCIN

Prof. S. Cohen, M.D. (Pharmacology)
 Dr. B. Buchanan (Computer Science)

PROTEIN CRYSTALLOGRAPHY MODELING

Dr. S. Freer (Chemistry, U.C. San Diego)
 Prof. E. Feigenbaum (Computer Science, Stanford)
 Dr. R. Engelmores (Computer Science, Stanford)

(NATIONAL)

COMPUTER MODEL OF DIAGNOSTIC LOGIC (DIALOG) -- U. of Pittsburgh
 Dr. H. Pople
 J. Myers, M.D.

HIGHER MENTAL FUNCTIONS MODELING -- UCLA

Kenneth M. Colby, M.D.

MEDICAL INFORMATION SYSTEMS LABORATORY (MISL) -- U. of Illinois

Dr. B. McCormick at Chicago Circle
 M. Goldberg, M.D.

RUTGERS RESEARCH RESOURCE COMPUTERS IN BIOMEDICINE -- Rutgers U.

Prof. S. Amarel

There are also a number of pilot-projects both at Stanford and nationally.

(DOCUMENTATION)

(PROGRAMS)

A list of the available on-line documentation of programs and general information files describing the SUMEX system and policies is available through running of the HELP program. A list of the hardcopy documentation available and procedures for obtaining copies

is contained in <DOC>A-LIST-OF-AVAILABLE-DOCUMENTATION.INFO.

(PROJECTS)

The following is a partial bibliography of research papers by the various projects. More complete bibliographies can be obtained by contacting the individual project leaders.

DENDRAL:

1. D. H. Smith, L. M. Masinter, and N. S. Sridharan, "Computer Representation and Manipulation of Chemical Information", W.T. Wipke, S. Heller, R. Feldmann, and E. Hyde, Eds., John Wiley and Sons, Inc., 1974, p. 287.
2. R. E. Carhart et. al., "Applications of Artificial Intelligence for Chemical Inference. XVIII. An Approach to Computer-Assisted Elucidation of Molecular Structure", J. Amer. Chem. Soc., in press, Sept. 1975.
3. Duffield et. al., "Applications of Artificial Intelligence for Chemical Inference. II. Interpretation of Low Resolution Mass Spectra of Ketones", J. Amer. Chem. Soc., 91, 2977, (1969).
4. R. E. Carhart, et. al., "Networking and a Collaborative Research Community: A Case Study Using the Dendral Programs", to appear in the Proceedings of the Amer. Chem. Soc., Aug. 1975.

HIGHER MENTAL FUNCTIONS MODELING:

1. W. S. Faight, K. M. Colby, R. C. Parkison, "The Interaction of Inferences, Affects, and Intentions in a Model of Paranoia", AIM-253, December 1974, Stanford AI Laboratory.
2. K. M. Colby, R. C. Parkison, B. Faight, "Pattern-Matching Rules for the Recognition of Natural Language Dialogue Expressions", AIM-234, Stanford Artificial Intelligence Laboratory, Stanford, California, June 1974.

MYCIN:

1. E. H. Shortliffe, F. Rhame, et. al., "MYCIN, A Computer Program Providing Antimicrobial Therapy Recommendations", Clinical Research, vol 23, p 107A (abstract) 1975.
2. E. H. Shortliffe, S. G. Axline, B. G. Buchanan, S. N. Cohen, "Design Considerations for a Program to Provide Consultations in Clinical Therapeutics". Presented at San Diego Biomedical Symposium 1974 (February 6-8, 1974).
3. E. H. Shortliffe, R. Davis, S. G. Axline, B. G. Buchanan, C. C. Green, and S. N. Cohen, "Computer-Based Consultations in Clinical Therapeutics: Explanation and Rule Acquisition Capabilities of the MYCIN System", to appear in Computers and

Biomedical Research, June 1975.

4. E. H. Shortliffe, "MYCIN, A Rule Based Computer Program...", STAN-CS-74-465, Computer Science Department, Stanford University, 1974.
5. E. H. Shortliffe, S. G. Axline, B. G. Buchanan, T. C. Merigan, and S. N. Cohen, "An Artificial Intelligence Program to Advise Physicians Regarding Antimicrobial Therapy", Computers and Biomedical Research, 6 (1973), 544-560.
6. E. H. Shortliffe and B. G. Buchanan, "A Model of Inexact Reasoning in Medicine", July 1974. To appear in Mathematical Biosciences.

RUTGERS RESEARCH RESOURCE ON COMPUTERS IN BIOMEDICINE:

- 1) C.A. Kulikowski, "Computer-Based Systems for Vision Care", Proc. IEEE Intercon, April 1975.
- 2) S. Amarel, "Computer-Based Modeling and Interpretation in Medicine and Psychology: The Rutgers Research Resource", Federation Proceedings, Vol. 33, No. 12.
- 3) C. A. Kulikowski, "Computer-Based Medical Consultation--A Representation of Treatment Strategies", Proc. Hawaii International Conf. on Systems Science, January 1974.
- 4) C. A. Kulikowski, "A System for Computer-Based Medical Consultation", Proc. National Computer Conference, Chicago, May 1974.
- 5) S. Amarel, "Inference of Programs from Sample Computations", Proceedings of NATO Advanced Study Institute on Computer Oriented Learning Processes, Bonas, France.
- 6) B. Bruce, "A Logic for Unknown Outcomes", Notre Dame Journal of Formal Logic.
- 7) S. Chokhani and C. A. Kulikowski, "Process Control Model for the Regulation of Intraocular Pressure and Glaucoma", Proc. IEEE Systems, Man Cybernetics Conf., Boston, November 1973.
- 8) C. F. Schmidt and J. D'Addamio, "A Model of the Common Sense Theory of Intension and Personal Causation", Proc. of the 3rd IJCAI, August 1973.
- 9) C. F. Srinivasan, "The Architecture of a Coherent Information System: A General Problem Solving System", Proc. of the 3rd IJCAI, August 1973.

APPENDIX H

AIM MANAGEMENT COMMITTEE MEMBERSHIP

The following are the membership lists of the various SUMEX-AIM management committees at the present time:

AIM EXECUTIVE COMMITTEE:

=====

LEDERBERG, Dr. Joshua (LEDERBERG) (Chairman)
Department of Genetics, S331
Stanford University Medical Center
Stanford, California 94305
(415) 497-5801

AMAREL, Dr. Saul (AMAREL)
Department of Computer Science
Rutgers University
New Brunswick, New Jersey 08903
(201) 932-3546

BAKER, Dr. William R., Jr. (BAKER) (Executive Secretary)
Biotechnology Resources Program
National Institutes of Health
Building 31, Room 5B25
9000 Rockville Pike
Bethesda, Maryland 20014
(301) 496-5411

LINDBERG, Dr. Donald (LINDBERG) (Adv Grp Member)
605 Lewis Hall
University of Missouri
Columbia, Missouri 65201
(314) 882-6966

AIM ADVISORY GROUP:

=====

LINDBERG, Dr. Donald (LINDBERG) (Chairman)
605 Lewis Hall
University of Missouri
Columbia, Missouri 65201
(314) 882-6966

AMAREL, Dr. Saul (AMAREL)
Department of Computer Science
Rutgers University
New Brunswick, New Jersey 08903
(201) 932-3546

BAKER, Dr. William R., Jr. (BAKER) (Executive Secretary)
Biotechnology Resources Program
National Institutes of Health
Building 31, Room 5B25
9000 Rockville Pike
Bethesda, Maryland 20014
(301) 496-5411

BOBROW, Dr. Daniel G. (BOBROW)
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, California 94304
(415) 494-4438

FEIGENBAUM, Dr. Edward (FEIGENBAUM)
Serra House
Department of Computer Science
Stanford University
Stanford, California 94305
(415) 497-4878

FELDMAN, Dr. Jerome (FELDMAN)
Department of Computer Science
University of Rochester
Rochester, New York
(716) 275-5671

LEDERBERG, Dr. Joshua (LEDERBERG) (Ex-officio)
Principal Investigator - SUMEX
Department of Genetics, S331
Stanford University Medical Center
Stanford, California 94305
(415) 497-5801

MILLER, Dr. George (GMILLER)
The Rockefeller University
1230 York Avenue
New York, New York 10021
(212) 360-1801

REDDY, Dr. D. R. (REDDY)

Department of Computer Science
 Carnegie-Mellon University
 Pittsburgh, Pennsylvania
 (412) 621-2600, Ext. 149

SAFIR, Dr. Aran (SAFIR)
 Department of Ophthalmology
 Mount Sinai School of Medicine
 City University of New York
 Fifth Avenue and 100th Street
 New York, New York 10029
 (212) 369-4721

STANFORD COMMUNITY ADVISORY COMMITTEE:

=====

LEDERBERG, Dr. Joshua (LEDERBERG) (Chairman)
 Principal Investigator - SUMEX
 Department of Genetics, S331
 Stanford University Medical Center
 Stanford, California 94305
 (415) 497-5801

COHEN, Stanley N., M.D. (COHEN)
 Division of Clinical Pharmacology, S169
 Stanford University Medical Center
 Stanford, California 94305
 (415) 497-5315

DJERASSI, Dr. Carl
 Department of Chemistry, Stauffer I-106
 Stanford University
 Stanford, California 94305
 (415) 497-2783

FEIGENBAUM, Dr. Edward (FEIGENBAUM)
 Serra House
 Department of Computer Science
 Stanford University
 Stanford, California 94305
 (415) 497-4878

LEVINTHAL, Dr. Elliott C. (LEVINTHAL)
 Department of Genetics, S047
 Stanford University Medical Center
 Stanford, California 94305
 (415) 497-5813

APPENDIX I

USER INFORMATION - GENERAL BROCHURE

Revised May 1976

The Stanford University Medical Experimental Computer (SUMEX) was established in January, 1974, to constitute the first national shared computing resource for medical research. An innovative effort to help biomedical scientists meet today's research requirements and to explore computer applications in many health fields ranging from basic research to bedside care, SUMEX is directed by Professor Joshua Lederberg, Chairman of Stanford's Department of Genetics. The project has been funded by a grant from the Division of Research Resources of the National Institutes of Health (Biotechnology Resources Program) for an initial term that expires in July, 1978.

At present, SUMEX consists of a powerful time-shared dual processor DEC-10 computer system. It is available to approved users throughout the United States over computer communications networks. The project's goals for its present 5-year term are: 1) the encouragement of applications of artificial intelligence in medicine (AIM), and 2) the managerial, administrative and technical demonstration of a nationally-shared technological resource for health research.

Such a resource offers scientists both a significant economic advantage in sharing expensive instrumentation and a greater opportunity to share ideas about their research. This is especially timely in computer science, a field whose intellectual and technological complexity tends to nurture relatively isolated research groups. Each group may then tend to pursue its own line of investigation with limited convergence on working programs available from others. In this respect, computer applications have demonstrated less mutual incremental progress from diverse sources than is typical of other sciences. The SUMEX-AIM project seeks to reduce these barriers to scientific cooperation in the field of artificial intelligence applied to health research.

ARTIFICIAL INTELLIGENCE

The term "artificial intelligence" (AI) refers to research efforts aimed at studying and mechanizing information-processing tasks that have required the application of some degree of human intelligence. Controversial speculations on how far this eventually may lead only distract from pragmatically useful applications of currently feasible art. The current emphasis in the field is to understand the underlying principles of a) efficient acquisition and utilization of material knowledge, and b) the programmed representation of conceptual abstractions

in reasoning, deductive, and problem-solving activities. At present, these are far more specialized and inflexible than human intellectual functions; however, in special domains they may be of comparable or greater power, e.g., in the solution of formal problems in organic chemistry or in the integral calculus.

AI systems are characterized by complex computational processes that are primarily non-numeric, e.g., graph-searching and symbolic pattern analysis. They involve procedures whose execution is controlled by diverse types and forms of knowledge about a given task domain, such as models, fragments of "advice", and systems of constraints or heuristic rules. Unlike conventional algorithms commonly based on a well-tailored method for a given task, AI procedures typically use a multiplicity of methods in a highly conditional manner--depending on the specific data in the task and a variety of sources of relevant information. The tangible objective of this approach is the practical development of computer programs which, using formal and informal knowledge together with mechanized hypothesis formation and problem-solving procedures, will offer more general and effective consultative tools for the clinician and medical scientist. Contexts in which experimental data already are acquired by machine may offer even richer opportunities.

Each authorized project in the SUMEX-AIM community is concerned in some way with the application of these principles to medical and health research problems. This type of "intelligent" assistance by computer program is perhaps best illustrated by the following brief descriptions of a selected sample of SUMEX-AIM projects.

DENDRAL

The DENDRAL project at Stanford, under the direction of Professor Lederberg, Genetics; Professor Edward Feigenbaum, Computer Science; and Professor Carl Djerassi, Chemistry, is aimed at assisting the biochemist in interpreting molecular structures from spectroscopic, physical and chemical information. In cases where the characteristic spectra of a compound are not catalogued in libraries, the DENDRAL programs carry out the rather laborious processes a chemist must go through to interpret the spectrum from "first principles". One of the DENDRAL programs, CONGEN (for CONstrained structure GENeration), is an interactive program designed to assist the chemist in the enumeration of structural isomers, based on inferences about structural features of an unknown compound. These inferences, whether obtained from physical, chemical or spectroscopic data, are supplied to CONGEN as structural fragments and related information, using a standard computer terminal connected to SUMEX-AIM. The program uses atoms and superatoms (non-overlapping structural fragments known to be present in the molecule) to construct structures; the procedure is restricted by a variety of constraints on desired and undesired substructures and ring systems. There is no direct algorithmic path available to determine such a molecular structure from the spectral data--only the inferential process of hypothesis generation and testing within the domain of reasonable solutions defined by a knowledge of organic and physical chemistry.

This process, as implemented in the computer, is a simplified example of the cycle of inductive hypothesis--deductive verification that is often taught as a model of the scientific method. (Whether this is a faithful description of contemporary science is arguable, and how it may be implemented in the human brain is unknown. Regardless, these are useful leads rather than absolute preconditions for the pragmatic improvement of mechanized intelligence for more efficient problem-solving.) The elaboration of these approaches with existing hardware and software technologies is the most promising approach to enhancing the application of computers to the vaguely structured problems that dominate our task domains.

A new pilot project, MOLGEN, has been motivated by the success of the DENDRAL effort. MOLGEN uses similar paradigms in an effort to mechanize experiment-planning in molecular genetics, particularly work on DNA structure and inter-species transfer being conducted in Professor Lederberg's laboratory. Whereas the DENDRAL goal was a hypothesis (i.e., a chemical structure) to explain a set of experimental data, MOLGEN begins with a stipulated DNA structure and seeks suggested experiment plans that could either falsify or validate the asserted structure. At present, this entails a substantial effort in representing existing knowledge of experimental techniques (i.e., enzyme specificities, electron-microscopy, electrophoresis) and the physical biochemistry of DNA.

THE RUTGERS PROJECT COMPUTERS IN BIOMEDICINE

Professor Saul Amarel, a Rutgers University computer scientist, directs several research efforts designed to introduce advanced methods in computer science--particularly in artificial intelligence and interactive data base systems--into specific areas of biomedical research.

For example, a group led by Professor Casimir Kulikowski is developing computer-based consultation systems for diseases of the eye in collaboration with Dr. Aran Safir, an ophthalmologist from the Mount Sinai School of Medicine. An important development in this area is the establishment of a national network of collaborators (called the ONET) for computer diagnosis and treatment of glaucoma. The computer system, which includes an elaborate pathophysiologic model of the disease, is being tested through the SUMEX-AIM network at five eye centers: Mount Sinai Hospital and Medical Center, New York; Washington University, St. Louis; The Johns Hopkins University, Baltimore; the University of Illinois at Chicago Circle; and the University of Miami. Glaucoma, in one form or another, affects 2% of all people over 40 years of age. It is a disease in which increased pressure within the eye may lead to irreparable optic nerve damage and blindness. The computer-based program has great potential for assisting clinicians and researchers in understanding the disease, diagnosing it more accurately and improving its treatment.

In another project, Professor Charles Schmidt, a social psychologist, is developing a theory of how people arrive at

interpretations of the social actions of others in collaboration with Professor N.S. Sridharan, a computer scientist. The theory will be tested in situations such as the psychiatric interview and the legal trial. The computer system which currently represents the theory is called "Believer". It includes a large body of statements about people's motivations and actions.

The Rutgers project includes, in addition, several fundamental studies in artificial intelligence and system design. These provide much of the support needed for the development of complex systems such as the glaucoma consultation and the "Believer" programs.

SIMULATION AND EVALUATION OF CHEMICAL SYNTHESIS

The development of new drugs and the study of how drug structure is related to biological activity depends on the chemist's ability to synthesize new molecules and modify existing structures, e.g., incorporating isotopic labels into biomolecular substrates. The Simulation and Evaluation of Chemical Synthesis (SECS) project, directed by Dr. Todd Wipke, Associate Professor of Chemistry at the University of California, Santa Cruz, is aimed at assisting the synthetic chemist in designing stereospecific syntheses of complex bio-organic molecules.

The molecule to be synthesized is presented to SECS using interactive computer graphics. The program studies the chemical graph and also 3-dimensional and electronic models of the molecule which it knows how to construct; then, using fundamental chemical principles, and various heuristics, it works backwards from the target to predict possible precursors which are one synthetic step away from the target. The chemist selects the precursors to be considered by the program for further analysis. Thus, SECS acts as a consultant, working with the chemist to form a chemist-computer team. The chemist helps guide the search and decides when to stop the analysis. Knowledge about chemical transformations is expressed directly by chemists in ALCHEM, an English-like language interpreted by SECS. Goals for further development of the project include generation of constraining strategies based on symmetry, steric and electronic considerations, and expansion of the chemical transform data base.

In addition to its on-going development on the SUMEX-TYMNET system, an experimental version of SECS is available over TELENET from First Data Corporation in Waltham, Massachusetts. SECS also runs on a Univac system at the University of Strasbourg, France, and on PDP-10's at the Universities of Darmstadt and Heidelberg, Germany. Feedback from this outside use of SECS spotlights areas for needed work and provides positive evidence of the usefulness of SECS as a tool in synthetic design.

MYCIN
Computer-based Consultation
in Clinical Therapeutics

Dr. Stanley Cohen, Professor and Head of the Division of Clinical Pharmacology at Stanford, directs this research in collaboration with Dr. Stanton Axline and with computer scientists interested in artificial intelligence and medical computing. The MYCIN system models the decision processes of medical experts, utilizing both clinical data and the judgmental knowledge of experts to provide physician nonspecialists with consultative advice regarding clinical therapeutics. Although initial research concerns the use of antimicrobial agents in the treatment of bacteremias, the system is being expanded to deal with the treatment of other infections.

The primary component of the system is the Consultation program which uses the physician's response to computer-generated questions about a patient to make deductions about the case. It then advises the physician on the infectious disease diagnosis and the recommended treatment for the patient. The utility and flexibility of this program are increased by three adjunct programs: 1) a Question-Answering program which answers questions about the system's knowledge base and about a specific consultation, 2) an Explanation program which justifies the consultative advice and explains the system's deduction process, and 3) a Knowledge Acquisition program which extends the knowledge base of the system through dialogue with an expert.

Goals for further development of the system include implementation and evaluation of the system in the clinical setting at the Stanford University and Palo Alto Veterans Administration Hospitals.

ACT
A Model of Human Cognition

The ACT Project is directed by Dr. John Anderson, Associate Professor of Psychology at Yale University. The ACT program provides a uniform set of theoretical mechanisms to model such aspects of human cognition as memory, inferential processes, language processing, and problem-solving. The knowledge base consists of two components. The propositional component is provided by an associative network encoding a set of known facts which provide the system's semantic memory. The procedural component consists of a set of productions which operate on the associative network. The production system used is considerably different than those in other currently available systems, e.g., Newell's PSG, and allows the system to operate on an associative network and to more accurately model certain aspects of human cognition.

ARTIFICIAL INTELLIGENCE METHODOLOGY APPLIED TO PROTEIN CRYSTALLOGRAPHY

Members of the artificial intelligence project at Stanford also are collaborating with Professor Joseph Kraut, Dr. Stephan Freer and other protein crystallographers at the University of California, San Diego. They are using the SUMEX-AIM facility as the central repository for programs, data and other information of common interest. The general objectives of the project are: 1) to identify critical tasks in protein structure elucidation which may benefit by the application of AI problem-solving techniques, and 2) to design and implement programs to perform those tasks.

Two principal task areas have been identified where collaboration is of practical and theoretical interest to both protein crystallographers and computer scientists working in AI: 1) interpreting a 3-dimensional electron density map, and 2) determining a plausible structure in the absence of phase information normally inferred from experimental isomorphous replacement data.

INTERNIST

The INTERNIST project, under the direction of Dr. Harry Pople and Dr. Jack Myers at the University of Pittsburgh, is a large-scale, computerized medical diagnostic system utilizing the methods and structures of artificial intelligence. Unlike most computer diagnostic programs, which are oriented to differential diagnosis in a rather limited area, the INTERNIST system deals with the general problem of diagnosis in internal medicine and currently accesses a medical data base encompassing approximately 50% of the major diseases in internal medicine.

MEDICAL INFORMATION SYSTEMS LABORATORY

The Medical Information Systems Laboratory (MISL) at the University of Illinois, Chicago Circle Campus, has been established under the direction of Dr. Bruce McCormick of the Department of Information Engineering, in collaboration with Dr. Morton Goldberg of the Department of Ophthalmology at the University of Illinois Medical Center.

The foremost goal of the resource is the exploration of artificial intelligence techniques in automated clinical decision-making in ophthalmology. Investigations into the construction of a data base in ophthalmology, and into distributed data base design, are ancillary goals. Incorporating reliable clinical information into the ophthalmology data base is a critical prerequisite to adequate clinical decision support. Core research concerns the exploration of inferential relationships between analytic data and the natural history of selected eye diseases, both in treated and untreated form.

MISL utilizes the computer facilities of the University of Illinois and the SUMEX-AIM network, providing the administrative structure for assembling the expertise of the collaborating departments. Serving as a bridge between diverse academic worlds, MISL promotes close involvement between engineering and medical faculty. The Illinois Eye and Ear Infirmary at the Medical Center, with a throughput of 50,000 patients per year, provides an ideal setting for the direct application of computer technology to real problems in clinical medicine.

SUMEX-AIM Management

A significant part of the SUMEX-AIM experiment has been the development of a management structure to maximize the utility of the computer capability for a national community.

Users of the SUMEX facility are divided for administrative purposes into two groups: 1) local, at Stanford University School of Medicine, and 2) national, elsewhere in the United States. As Principal Investigator for the SUMEX grant, Dr. Lederberg reviews Stanford medical school projects with the assistance of a local advisory committee. National users may gain access to the facility through an advisory panel for a national program in Artificial Intelligence in Medicine (AIM). The AIM Advisory Group consists of members-at-large of the AI and medical communities, facility users and the Principal Investigator of SUMEX as an ex-officio member. A representative of the National Institutes of Health-Biotechnology Resources Program (NIH-BRP) serves as Executive Secretary.

The SUMEX-AIM computing resource is allocated initially to qualified users without fee. This, of course, entails a careful review of the merits and priorities of proposed applications. At the direction of the Advisory Group, expenses related to communications and transportation to allow specific users to visit the facility also may be covered.

SUMEX-AIM is aware of the necessity of making the central facility available for trial use by potential users and collaborators. A GUEST mechanism has been established for those who have an indicated requirement for brief access to certain programs. Those who have been given an appropriate telephone number and login procedure can dial up SUMEX-AIM to exercise these programs on a trial basis. A specific objective of many user projects is the demonstration of their programs for the benefit of a highly dispersed national community.

USER QUALIFICATIONS

The SUMEX-AIM facility is a community effort, not merely a machine service. Applications for membership are judged on the basis of the following criteria:

- 1) The scientific interest and merit of the proposed research and its relevance to the health research missions of the NIH.
- 2) The congruence of research needs and goals to the AI functions of SUMEX-AIM as opposed to other computing alternatives.
- 3) The user's prospective contributions and role in the community, with respect to computer science, e.g., developing and sharing new systems or applications programs, sharing use of special hardware, etc.
- 4) The user's potential for substantive scientific cooperation with the community, e.g., to share expert knowledge in relevant scientific specialties.
- 5) The quantitative demands for specific elements of the SUMEX-AIM resource, taking account of both mean and ceiling requirements.

FACILITY INFORMATION

The computer facility, consists of dual DEC Model KI-10 CPU's running under a locally-developed dual processor TENEX operating system. It has 256K words (36-bit) of high-speed memory, 1.6M words of swapping storage, 70M words of disk storage, two 9-track 800 bpi industry-compatible tape units, a dual DEC-tape unit, a line printer, and communications-network interfaces providing user terminal access. SUMEX may be accessed by local telephone lines, through the TYMNET and as a host over the ARPANET communications network.

Program (software) support will evolve from the basic system as dictated by the research goals and needs of the user. Initially, available programs include a variety of TENEX user, utility and text editor programs. Major user languages include INTERLISP, SNOBOL, SAIL, FORTRAN-10, BLISS-10, BASIC, Macro-10, OMNIGRAPH and MLAB.

POTENTIAL USERS

Potential users seeking further information are invited to write:

Elliott Levinthal, Ph.D.
AIM User Liaison
SUMEX-AIM Computer Project
c/o Department of Genetics, S047
Stanford University Medical Center
Stanford, California 94305
Telephone: (415) 497-5813

Procedures for access to SUMEX-AIM are governed by the:

Biotechnology Resources Program
Division of Research Resources
National Institutes of Health
Building 31, Room 5B19
Bethesda, Maryland 20014

APPENDIX J

GUIDELINES FOR PROSPECTIVE USERS

SUMEX-AIM RESOURCE
INFORMATION FOR POTENTIAL USERS

National users may gain access to the facility resources through an advisory panel for a national program in Artificial Intelligence in Medicine (AIM). The AIM Advisory Group consists of members-at-large of the AI and medical communities, facility users and the Principal Investigator of SUMEX as an ex-officio member. A representative of the National Institutes of Health-Biotechnology Resources Program (NIH-BRP) serves as Executive Secretary.

Under its enabling 5-year grant, the SUMEX-AIM computing resource is allocated to qualified users without fee. This, of course, entails a careful review of the merits and priorities of proposed applications. At the direction of the Advisory Group, expenses related to communications and transportation to allow specific users to visit the facility also may be covered.

USER QUALIFICATIONS

The SUMEX-AIM facility is a community effort, not merely a machine service. Applications for membership are judged on the basis of the following criteria:

- 1) The scientific interest and merit of the proposed research and its relevance to the health research missions of the NIH.
- 2) The congruence of research needs and goals to the AI functions of SUMEX-AIM as opposed to other computing alternatives.
- 3) The user's prospective contributions and role in the community, with respect to computer science, e.g., developing and sharing new systems or applications programs, sharing use of special hardware, etc.
- 4) The user's potential for substantive scientific cooperation with the community, e.g., to share expert knowledge in relevant scientific specialties.
- 5) The quantitative demands for specific elements of the SUMEX-AIM resource, taking account of both mean and ceiling requirements.

In many respects, this requires a different kind of information for

judgment of proposals than that required for routine grant applications seeking monetary funding support. Information furnished by users also is indispensable to the SUMEX staff in conducting their planning, reporting and operational functions.

The following questionnaire encompasses the main issues concerning the Advisory Group. However, this should neither obstruct clear and imaginative presentation nor restrict format of the application. The potential user should prepare a statement in his own words using previously published material or other documents where applicable. In this respect, the questionnaire may be most useful as a checklist and reference for finding in other documentation the most cogent replies to the questions raised.

For users mounting complex and especially non-standard systems, the decision to affiliate with SUMEX may entail a heavy investment that would be at risk if the arrangement were suddenly terminated. The Advisory Group endeavors to follow a responsible and sensitive policy along these lines--one reason for cautious deliberation; and even in the harshest contingencies, it will make every effort to facilitate graceful entry and departure of qualified users. Conversely, it must have credible information about thoughtful plans for long-term requirements including eventual alternatives to SUMEX-AIM. SUMEX-AIM is a research resource, not an operational vehicle for health care. Many programs are expected to be investigated, developed and demonstrated on SUMEX-AIM with spinoffs for practical implementation on other systems. In some cases, the size, scope and probable validation of clinical trials would preclude their being undertaken on SUMEX-AIM as now constituted. Please be as explicit as possible in your plans for such outcomes.

Applicants, therefore, should submit:

- 1) One to two-page outline of the proposal.
- 2) Response to questionnaire, cross-referenced to supporting documents where applicable.
- 3) Supporting documents.
- 4) List of submitted materials, cross-referenced.

We would welcome a draft (2 copies) of your submission for informal comment if you so desire. However, for formal consideration by the SUMEX-AIM Advisory Group, please submit 13 copies of the material requested above in final form.

Elliott Levinthal, Ph.D.
AIM User Liaison
SUMEX-AIM Computer Project
c/o Department of Genetics, S047
Stanford University Medical Center
Stanford, California 94305
Telephone: (415) 497-5813

May, 1976

SUMEX-AIM RESOURCE

QUESTIONNAIRE FOR POTENTIAL USERS

Please provide either a brief reply to the following or cite supporting documents.

A) MEDICAL AND COMPUTER SCIENCE GOALS

- 1) Describe the proposed research to be undertaken on the SUMEX-AIM resource.
- 2) How is this research presently supported? Please identify application and award statements in which the contingency of SUMEX-AIM availability is indicated. What is the current status of any application for grant support of related research by any federal agency? Please note if you have received notification of any disapproval or approval, pending funding, within the past three years. Budgetary information should be furnished where it concerns operating costs and personnel for computing support. Please furnish any contextual information concerning previous evaluation of your research plans by other scientific review groups.
- 3) What is the relevance of your research to the AI approach of SUMEX-AIM as opposed to other computing alternatives?

B) COLLABORATIVE COMMUNITY BUILDING

- 1) Will the programs designed in your research efforts have some possible general application to problems analogous to that research?
- 2) What application programs already publically available can you use in your research? Are these available on SUMEX-AIM or elsewhere?
- 3) What opportunities or difficulties do you anticipate with regard to making available your programs to other collaborators within a reasonable interval of publication of your work?
- 4) Are you interested in discussing with the SUMEX staff possible ways in which other artificial-intelligence research capabilities might interrelate with your work?
- 5) If approved as a user, would you advise us regarding collaborative opportunities similar to yours with other investigators in your field?

C) HARDWARE AND SOFTWARE REQUIREMENTS

- 1) What computer facilities are you now using in connection with your research or do you have available at your institution? In what respect do these not meet your research requirements?
- 2) What languages do you either use or wish to use? Will your research require the addition of major system programs or languages to the system? Will you maintain them? If you are committed to systems not now maintained at SUMEX, what effort would be required for conversion to and maintenance on the PDP-10 - TENEX system? What are the merits of the alternative plan of converting your application programs to one of the already available standards? Would the latter facilitate the objectives of Part B), Collaborative Community Building?
- 3) Can you estimate your requirements for CPU utilization and disk space? What time of day will your CPU utilization occur? Would it be convenient or possible for you to use the system during off-peak periods? Please indicate (as best you can) the basis for these estimates and the consequences of various levels of restriction or relaxation of access to different resources. SUMEX-AIM's tangible resources can be measured in terms of:
 - a) CPU cycles.
 - b) Connect time and communications.
 - c) User terminals (In special cases these may be supported by SUMEX-AIM.).
 - d) Disk space.
 - e) Off-line media-printer outputs, tapes (At most, limited quantities to be mailed.).

Can you estimate your requirements? With respect to a) and b), there are loading problems during the daily cycle.--Can you indicate the relative utility of prime-time (0900-1600 PST) vs. off-peak access?

- 4) What are your communication plans (TYMNET, ARPANET, other)? How will your communication and terminal costs be met? See following note concerning network connections to SUMEX-AIM.
- 5) If this is a development project, please indicate your long-term plans for software implementation in an applied context keeping in mind the research mission of SUMEX-AIM.

Our procedures are still evolving, and we welcome your suggestions

about this framework for exchanging information. Needless to say, each question should be qualified a) "insofar as relevant to your proposal", and b) "to the extent of available information".

Please do not force a reply to a question that seems inappropriate. We prefer that you label it as such so that it can be dealt with properly in future dialogue.

Above all, we are eager to work with potential users in any way that would help minimize bureaucratic burdens and still permit a responsible regard for our accountability both to the NIH and the public. Please do not hesitate to address the substance of these requirements in the format most applicable to you.

NETWORK CONNECTIONS TO SUMEX-AIM

TYMNET

Attached is a list of available TYMNET nodes and associated telephone numbers. The cost to users of using TYMNET is the telephone charge from user location to the nearest TYMNET node. This is available only for communication to SUMEX-AIM and not for other facilities that may be connected to TYMNET. In some cases, there are "foreign exchanges" set up by users. These may offer less expensive communication. Details of these possibilities can best be learned by calling the nearest TYMNET node. The telephone company can provide information on comparative costs of leased lines, toll charges, etc. The initial capital investment for TYMNET installation as well as login and hourly charges is provided by SUMEX-AIM. Standard usage charges on TYMNET are approximately \$3/connect-hour.

ARPANET

SUMEX-AIM is connected to the ARPANET. Our name is SUMEX-AIM; our nickname is AIM. We support the new TELNET protocol. Our network address is decimal 56, octal 70. This provides convenient access for ARPANET Hosts and Associates and those who have accounts with ARPANET.