

1.3.2.5 SYSTEM RELIABILITY AND BACKUP

System reliability has remained high over the past years; excellent under stable hardware and software conditions and degrading temporarily during debugging and development periods and during periods of difficult hardware problems. In general we take the system down for approximately 50 hours per month for scheduled hardware maintenance, file backup, and other maintenance. In addition we average from 10 to 15 hours per month in unscheduled downtime. During particularly difficult hardware or software difficulties we must absorb substantially more downtime.

1.3.2.6 PROGRAMMING LANGUAGES

Over the past years 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 ALGOL-like system (MAINSAIL).

LISP Efficiency:

There has been an on-going debate among a number of projects over the best language to choose for developmental implementation of the various AI programs. The key issues include ease and flexibility of conceptual representation of program functions and objects, interactive debugging support, efficiency, and exportability. To date the predominant language choice for AIM research has been LISP and more particularly INTERLISP. 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. We recently attempted an evaluation of INTERLISP and ILISP including the relative efficiencies of the two languages and the level of assistance the language systems provide the user in developing programs. 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 factor of 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 reimplementations 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.

MAINSAIL Progress

SUMEX, in its role as a nationally shared computer resource, is an appropriate vehicle for the development of 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.

The MAINSAIL (3) project has three closely related goals:

- 1) Provide an integrated set of tools for the creation of efficient portable software on a variety of computer systems, and provide support and continued development of these tools in a form compatible across all implementations.
- 2) Study innovative approaches to portability, both hardware and software, and develop such approaches into effective tools.
- 3) Promote the development and distribution of portable software, advise and assist in its design, and evaluate its applicability.

By portable software we mean computer programs which may be executed on a variety of machines with few, if any, alterations. MAINSAIL itself will provide the initial example of portable software, since all of the system is written in the MAINSAIL language except for those parts which are determined by the host environment (hardware, instruction set, operating system, etc.). Even these parts are embedded within MAINSAIL.

(3) The MAINSAIL (MACHINE-INdependent SAIL) 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. A summary of some of the features of the MAINSAIL language and their relationship to other languages is given in Appendix III on page 231 (see Book II).

There is a key distinction between MAINSAIL's approach to portability and the "classical" approach characterized by languages such as FORTRAN, ALGOL, LISP, COBOL and BASIC. These languages attempt to adhere to a single syntax standard which is separately implemented for each different computer system. Invariably these implementations have differences which preclude the creation of a program which is accepted by all. It is difficult, if not impossible, to define a language standard which is unambiguous and at the same time sufficiently comprehensible to provide the basis for compatible implementations. Furthermore, many implementors yield to the temptation to provide "enhancements" to the standard which immediately introduces machine and system dependencies.

MAINSAIL, on the other hand, provides a single system (written primarily in itself) which is employed at every site. This is made possible by its ability to compile itself into code for a variety of machines. Only the compiler's code generators and the runtime operating-system interfaces need be rewritten for each implementation. These parts of MAINSAIL are at a level which has already been defined by the machine-independent parts, and do not affect the language from the user's viewpoint. Thus the "language standard" has been reduced to a "semantic standard" which is surrounded by machine-independent software.

It remains to be seen whether the temptation to augment the language with machine-dependencies (for purposes of ultimate efficiency or to take advantage of particular local system features) can be overcome. Herein also lies the biggest "price" to be paid for exportability. The code emitted from the MAINSAIL compiler can be (and is, based on tests to date) at least as efficient as that from many machine-dependent compilers. On the other hand, special machine or operating system features that cannot be uniformly implemented may provide local optimizations at the cost of exportability or vice versa. We cannot effectively measure the extent of this cost at this stage.

DEVELOPMENT APPROACH

We do not underestimate the difficulty in obtaining the cooperation of a community which will span a wide variety of applications and hardware/software systems. If MAINSAIL is to obtain widespread use, it is crucial that it have an effective and credible base of support. The initial parts of MAINSAIL are just about ready for limited distribution. We want to maintain close supervision of this distribution, and insure that systems labelled as MAINSAIL are not altered without our approval. In this regard we are pursuing legal channels to safeguard the integrity of MAINSAIL software. We plan to take MAINSAIL through an orderly progression of development, and to avoid casual distribution with no provision for a solid base of maintenance and future growth.

REVIEW OF PROGRESS TO DATE

MAINSAIL has been under development for almost three years now. Beginning with an initial goal of converting the PDP-10 SAIL compiler to generate code for a PDP-11, several versions had been implemented on a PDP-10 and a PDP-11, and the groundwork had been laid for extending the system to a wider variety of machines. The current version was begun in August of 1976.

Early versions of MAINSAIL attempted to maintain close compatibility with the original SAIL, but in surveying a wider variety of machines (especially mini-computers), we concluded that this compatibility could be maintained only at the expense of portability. It was felt that MAINSAIL could contribute more by providing a truly portable system. Thus we began redesigning MAINSAIL, rebuilding from previous implementations. This effort has resulted in a new version which is still under development, and is now being tested on several systems.

Initial implementations of the current design are for DEC PDP-10's with the TENEX operating system and with the TOPS-10 operating system. The TENEX version is being tested at SUMEX and has been installed at one other TENEX site (Stanford - IMSSS). The TOPS-10 version was developed at SUMEX by using TENEX facilities which provide compatibility with TOPS-10. The Rutgers University PDP-10 facility was chosen for external testing since it is a standard TOPS-10 system, and can be accessed from SUMEX over a network. MAINSAIL is now undergoing preliminary testing there. A modified TOPS-10 version has been set up on the Stanford AI-lab's PDP-10, but also has not been open to general use.

Little additional work will be necessary to make the TENEX version execute on a DECSYSTEM-20 since TOPS-20 is derived from TENEX. However, some time will be needed to take full advantage of the extended instruction set of the KL-10. Two sites are available for TOPS-20 development: the LOTS facility at Stanford; and a machine at SRI, close to Stanford and accessible over a network. Both of these sites have expressed an interest in using MAINSAIL.

The PDP-11 has been chosen as the first mini-computer to be implemented. Code generators have been written for it but not debugged. Several variants of these code generators will be necessary to cover the full PDP-11 family.

MAINSAIL interfaces to three PDP-11 operating systems (RT-11, RSX-11 and UNIX) are now under development. All of these operating systems are available to the MAINSAIL project on PDP-11's at Stanford. RT-11 will be the first to be implemented. The mix of instruction sets, operating systems and configurations will be a good test of MAINSAIL's ability to provide a compatible implementation, even across this one family of computers. We expect the PDP-11 systems to be operational by this summer.

1.3.2.7 STANFORD AI HANDBOOK PROJECT

The AI Handbook is a compendium of short articles (3-5 pages each) about the projects, ideas, problems and techniques that make up the field of Artificial Intelligence. Over 150 articles have been drafted by researchers and students in the field, on topics ranging in depth from "Augmented Transaction Networks" (ATN's) to "An Overview of Natural Language Research", and covering the entire breadth of AI research: search, robotics, speech understanding, real-world applications, etc. An outline of the current contents of the handbook is given in Appendix II on page 225 (see Book II).

During the Spring of 1976 the final push for drafting new articles was completed, with some 60 articles produced by students during that quarter. Since then the process has begun of rewriting the various chapters of the Handbook to produce coherent manuscripts from the original work of five to ten authors. This effort involves rewriting articles for accuracy and completeness as well as integrating the 15 to 25 articles in a section into an editorially uniform and readable document. An editor has been added to the project team who will be responsible for maintaining a consistent format and style in the Handbook.

When completed, each chapter will be reviewed by experts in the appropriate research area before it is released to the public. At present, the chapter on Natural Language research is completed and being reviewed, and we expect that the sections on Search, Speech Understanding, Representation of Knowledge, and Automatic Programming will be completed during the next two months. During the Fall of 1977 the first seven chapters of the handbook will be published in preliminary form. Meanwhile, the handbook is already available to cooperative experts and critics on-line via the SUMEX-AIM network connections. We are considering maintaining the handbook on-line, with occasional hard-copy editions, and believe this method of "publication" may be a prototype for other encyclopedic monographs.

1.3.2.8 USER SOFTWARE AND INTRA-COMMUNITY COMMUNICATION

In addition to the system and language software development efforts of SUMEX, we have assembled or developed where necessary a broad range of utilities and user software. These include operational aids, statistics packages, DEC-supplied programs, improvements to the TOPS-10 emulator, text editors, text search programs, file space management programs, graphics support, a batch program execution monitor, text formatting and justification assistance, and magnetic tape conversion aids. We have also developed a number of user information assistance programs such as a "WHOIS" facility to recover names and affiliations of users and a "HELP" facility to locate on-line documentation of interest through key word searches.

Of major importance for our community effort is the set of tools for inter-user communications. We have enhanced the message sending and manipulation programs to better integrate text editing facilities for easier message preparation and reading. We have also developed a unique "bulletin board" system to deal with informal notes, thereby bridging a functional gap between formal system documents and private messages communications between individual users. The bulletin board system provides an informal and dynamic base for information about system facilities, lore, bugs, etc. or can provide a means for intra-project communication and coordination.

The system has been in operation for more than one year and has been exported to IMSSS (Stanford's other TENEX site) and USC-ECL. We have also proposed that the next generation of ARPANET information services provide for bulletin board-like facilities. At SUMEX-AIM there are 10 bulletin boards, 8 of which are project-specific. The main system bulletin board currently contains more than 140 bulletins under 85 topics covering system status announcements,

explanations of recent crashes, hardware troubles and monitor upgrades, new developments, bugs, and little-documented features of our programming languages and utilities. Project bulletin boards have been used for notices and minutes of meetings, references to and abstracts of papers, coordination of on-going developments, vacation schedules, documentation and announcements of various kinds.

Current Bulletin Board features include:

Multiple bulletin boards (public, private, general, specific, etc.).

Topics and subtopics (separated by periods) may be nested to any depth.

Expire dates for each bulletin, after which they are removed automatically.

Interest-list-of-topics for each user allows him to be notified about new bulletins he is interested in and to ignore others.

Users notified when new bulletins arrive, by running BBCHECK (the bulletin-board MAIL CHECK) or by mail.

Help and browsing facilitated in a variety of ways (? can be typed anywhere, general and command-specific help provided).

Command structure modelled after the TENEX EXEC, with conscious attention to human-engineering.

Companion program BBREAD is a bulletin-board READMAIL.

Companion program BBNEWS types out a directory listing of any new bulletins.

1.3.2.9 DOCUMENTATION AND EDUCATION

We have spent considerable effort to develop, maintain, and facilitate access to our documentation so as to accurately reflect available software. The HELP and Bulletin Board systems have been important in this effort. We have limited manpower for user assistance. In general, users are responsible for their own software development and maintenance. The SUMEX staff, however, (including Lederberg and Rindfleisch) share the responsibilities for system level assistance to users, tracking down bugs, reviewing user suggestions, etc. The terminal linking facilities of TENEX have been valuable tools to assist remote user groups and also for system users to communicate with each other. With the recent initial release of the MAINSAIL system on selected machines, we are becoming increasingly involved in describing MAINSAIL and advising user projects in its possible applications.

1.3.2.10 SOFTWARE COMPATIBILITY AND SHARING

At SUMEX-AIM we firmly believe in importing rather than reinventing software where possible. 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.

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. Interactions have included mutual backup support, hardware configuration experiments, operating system enhancements, utility or language software, and user project collaborations. We have been able to import many new pieces of software and improvements to existing ones in this way. Examples of imported software include the message manipulation program MSG, TENEX SAIL, TENEX SOS, INTERLISP, the RECORD program, ARPANET host tables, and many others. Reciprocally, we have exported our contributions such as the drum page migration system, KI-10 page table efficiency improvements, GTJFN enhancements, PUB macro files, the bulletin board system, SNDMSG enhancements, our BATCH monitor, etc. The most recent example of this cooperative use of networks is in the preliminary export of MAINSAIL.

1.3.2.11 RESOURCE MANAGEMENT

PHILOSOPHY OF MANAGEMENT

The tidiest way to administer a national resource would be by subcontract to a fee-compensated, neutral agent. This would still have to involve a governing body that could speak to the technical and quality-control interests of the served constituency. Appropriate in some circumstances, this model would separate the administration of a resource from active research and development. An approach expected to foster greater creativity is to couple the resource with an active user-center. This of course can lead to manifest conflicts of interest that must be addressed and avoided if the resource is to be fairly available on a regional or national basis.

As indicated in the introduction, our proposal for the latter approach was followed by searching negotiations over a management plan that would be sensitive to these considerations. The bureaucratic procedures, much as they have to be

spelled out, are almost the last items that need to be specified for such a plan. Far more important is a charter that spells out the underlying objectives and responsibilities of the program, and which establishes incentives, resources, and obligations for proper performance. We believe the plan that was negotiated and implemented has all of these ingredients, and has made the design of the procedural framework a matter of simple common-sense logic from these premises. It will be plain that the convergence of local self-interest, and peer and contractual responsibility offers the best assurance that the programmatic goals will be respected, and simplifies the tasks of surveillance and accountability.

The self-interest part of this equation stems from our original motivation in requesting the resource: the need for specialized computing facilities to support intense, interdisciplinary studies in applications of AI at Stanford University Medical School. Comprising several departments (Genetics, Medicine, Computer Science and Chemistry), and interwoven projects (e.g., DENDRAL, Heuristic Programming, MYCIN, MOLGEN) and principal faculty (Professors Lederberg, Feigenbaum, Djerassi, Cohen, and Buchanan), a substantial body of research that has progressed and evolved over many years would be sacrificed if such a resource were not available. Successful, stable collaborations of this scope are not readily found. This history both depends upon and contributes to the doctrine of resource-sharing that underlies the SUMEX-AIM effort.

One premise of the management plan was therefore the charter allocation of half the user-available capacity of the SUMEX facility to the Stanford complex of projects, subject to a local committee chaired by Professor Lederberg.

The acceptance of this principle clearly defines the local benefit of the resource, minimizes anxiety and conflict-of-interest, and en suite enables the local group to respond quite objectively to the allocations that are made by an Executive Committee for the "national" or non-Stanford aliquot (see "Executive and Advisory Committee Organization" below). Another important contribution to the success of the plan is the welcome participation of an NIH-BRP representative on the Executive Committee. What would be inappropriate meddling, in the conduct of a narrower research project funded by NIH, is a communication channel and source of detached judgment that has been invaluable in expediting the innumerable decisions about which NIH must and should be consulted in the week-to-week business of the resource. The efficacy of this principle, as is appropriate to acknowledge here, has been validated and enhanced by the style and energy that Dr. William Baker has brought to this task.

That the "national" community should be conscientiously cultivated for the most efficacious use of its aliquot, and that further growth of facilities should in due course be distributed, are further inferences from the charter principles.

Finally, the recognition in the charter that SUMEX-AIM was not merely a retail-store for computer cycles, but the means of building a community, was a necessary basis for the morale of the whole operation. Some of these matters were addressed further in the section on SIGNIFICANCE (see Section 1.2 on page 4). The remainder of this section will now speak to the way in which these responsibilities are handled bureaucratically.

ORGANIZATION AND PROCEDURES

The SUMEX-AIM resource is administered within the Genetics Department of the Stanford University Medical School, Professor Lederberg's "main office", though he also holds appointments in the Computer Science Dept. and the Human Biology program. Its mission, locally and nationally, 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. 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. 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 6 on page 41 in Book II). We have pondered the possibilities of a fee-for-service approach to allocation of the resource. We believe that this would be inappropriate for an experimental system of such national scope, whose pricing structure would have to be revised almost on a week-to-week basis to fairly respond to evolutionary changes in the system. This would also pose problems of accountability for the transfer of funds from one institution to another. Our present policy of non-monetary allocation control, which we propose to continue for the next term, of course accentuates our responsibility for the careful selection of projects with high scientific and community merit.

EXECUTIVE AND ADVISORY COMMITTEE ORGANIZATION

As the SUMEX-AIM project is a multilateral 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 initial resource grant was awarded, the available facility capacity is allocated 40% to Stanford Medical School projects, 40% to national projects, and 20% to common 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 V (see Book II).

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 (e.g., hardware upgrades, MAINSAIL development priorities, etc.). The Executive Committee oversees the planning and implementation of the AIM Workshop series currently implemented under Prof. S. Amarel of Rutgers University and assures coordination with other AIM activities as well. The committee will play a key role in assessing the possible need for additional future AIM community computing resources and in deciding the optimal placement and management of such facilities. The current membership of the Executive committee is listed in Appendix V (see Book II).

Reporting to the Executive Committee, an Advisory Group represents the interests of 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 V (see Book II).

These committees have actively functioned in support of the resource. Except for the meetings held during the AIM workshops, the committees have met by telephone conference owing to the size of the groups and to save the time and expense of personal travel to meet face to face. These telephone meetings, 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. Other solicitations of advice requiring review of sizable written proposals are done by mail.

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.

NEW PROJECT RECRUITING

The SUMEX-AIM resource has been announced through a variety of media as well as by correspondence, contacts of NIH-BRP with a variety of prospective grantees who use computers, and contacts by our own staff and committee members. The number of formal projects that have been admitted to SUMEX has more than doubled since the start of the project; 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 a brochure (see Appendix VI in Book II) 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 VII in Book II). 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 their application for funding support and in parallel formal application for access to SUMEX. Pilot projects are approved for access for limited periods of time after preliminary review by the Stanford or AIM Advisory Group as appropriate to the origin of the project.

These contacts have sometimes done much more than provide support for already-formulated programs. For example, Prof. Feigenbaum's group at Stanford has initiated a major collaborative effort with Dr. Osborn's group at the Institutes of Medical Sciences in San Francisco. This project in "Pulmonary Function Monitoring and Ventilator Management - PUFF/VM" (see Section 6.4.6 on

page 197 in Book II) originated as a pilot request to use MLAB in a small way for modeling. Subsequently the AI potentialities of this domain were recognized by Feigenbaum, Nii, and Osborn who have submitted a joint proposal to NIH and have a pilot status at present.

The following lists the fully authorized projects currently comprising the SUMEX-AIM community (see Section 6 in Book II for more detailed descriptions). The nucleus of five projects that were authorized at the initial funding of the resource in December 1973 are marked by "<*>".

National -

- 1) Acquisition of Cognitive Procedures (ACT); Dr. J. Anderson (Yale University)
- <*> 2) Higher Mental Functions Project; K. Colby, M.D. (University of California at Los Angeles)
- 3) INTERNIST Project; J. Myers, M.D. and Dr. H. Pople (University of Pittsburgh)
- 4) Medical Information Systems Laboratory (MISL); J. Wilensky, M.D. and Dr. B. McCormick (University of Illinois at Chicago Circle)
- <*> 5) Rutgers Computers in Biomedicine; Dr. S. Amarel (Rutgers University)
- 6) Chemical Synthesis Project (SECS); Dr. T. Wipke (University of California at Santa Cruz)

Stanford -

- <*> 1) DENDRAL Project; Drs. C. Djerassi, J. Lederberg, and E. Feigenbaum
- 2) Large Multi-processor Arrays (HYDROID); Dr. G. Wiederhold
- 3) Molecular Genetics Project (MOLGEN); Drs. J. Lederberg, E. Feigenbaum, and N. Martin
- <*> 4) MYCIN Project; S. Cohen, M.D. and Dr. B. Buchanan
- <*> 5) Protein Structure Modelling; Drs. J. Kraut and S. Freer (University of California at San Diego) and E. Feigenbaum (Stanford)

As an additional aid to new projects or collaborators with existing projects, we provide a limited amount of funds for use 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.

STANFORD COMMUNITY BUILDING

The Stanford community has undertaken several internal efforts to encourage interactions and sharing between the projects centered here. 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 as reported earlier. An outline of the material being prepared can be found in Appendix II on page 225 (see Book II). Several examples of completed articles are given in Appendix I on page 202 (see Book II).

A second community-building effort was a mini-conference on AI 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. Weekly informal lunch meetings (SIGLUNCH) are also held between community members to discuss general AI topics, concerns and progress of individual projects, or system problems as appropriate as well as having a number of outside invited speakers.

AIM WORKSHOP SUPPORT

The Rutgers Computers in Biomedicine resource (under Dr. Saul Amarel) has organized a series of workshops devoted to a range of topics related to artificial intelligence research, medical needs, and resource sharing policies within NIH. Meetings have been held for the past two years at Rutgers and another is planned for this summer. The SUMEX facility has acted as a prime computing base for the workshop demonstrations. We expect to continue this support for future workshops. 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 continue to use this experience to guide the community building aspects of SUMEX-AIM.

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.

We have recently implemented system scheduling controls to attempt to maintain the 40:40:20 balance in terms of CPU utilization (see page 18). The initial complement of user projects justifying the SUMEX resource was centered to a large extent at Stanford. Over the first term of the SUMEX grant, a substantial growth in the number of national projects was realized. During the same time the Stanford group of projects has matured as well and in practice the 40:40 split between Stanford and non-Stanford projects is not ideally realized (see Figure 8 on page 43 and the tables of recent project usage on page 45). Our job scheduling controls bias the allocation of CPU time based on percent time consumed relative to the time allocated over the 40:40:20 community split. The controls are "soft" however in that they do not waste computer cycles if users below their allocated percentages are not on the system to consume the cycles. The operating disparity in CPU use to date reflects a substantial difference in demand between the Stanford community and the developing national projects, rather than inequity of access. For example, the Stanford utilization is spread over a large part of the 24-hour cycle, while national-AIM users tend to be more sensitive to local prime-time constraints. (The 3-hour time-zone phase shift across the continent is of substantial help in load-balancing.) For the present, we propose to continue our policy of "soft" allocation enforcement for the fair split of resource capacity. If necessary to assure proper apportionment, we can implement a pie-slice reservation system to more rigidly control the allocations.

Our system also categorizes 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 have had relatively little abuse compared to what other network sites have experienced, perhaps on account of the personal attention that senior staff gives to the logon records, and to other security measures. 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.

1.3.2.12 SUMMARY OF RESOURCE USAGE

The following data give an overview of SUMEX-AIM resource usage. There are five sub-sections containing data respectively for 1) monthly CPU time consumed, 2) resource usage by community (AIM and Stanford), 3) resource usage by project, 4) recent diurnal loading data, and 5) Network usage data.

MONTHLY CPU TIME CONSUMED

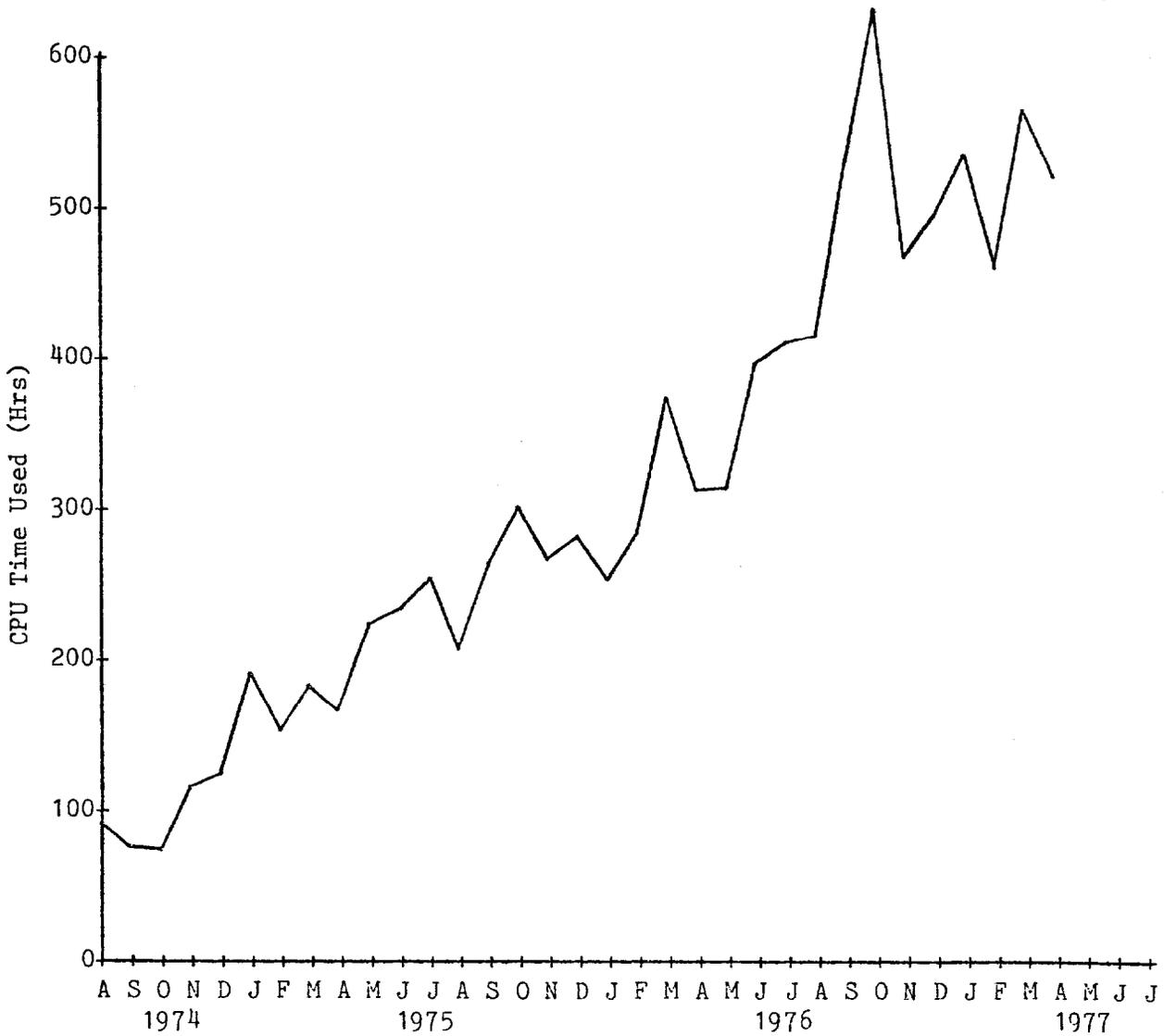


Figure 7. Monthly CPU Time Consumed

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 common systems development efforts. As defined in the resource management plan approved by BRP at the start of the project, the available resource in terms of CPU capacity and file space will be divided between these communities as follows:

Stanford	40%
AIM	40%
Staff	20%

The "available" resources to be divided up in this way are those remaining after various monitor and community-wide functions are accounted for. These include such things as job scheduling, overhead, network service, file space for subsystems and documentation, etc.

The monthly usage of CPU and file space resources for each of these three communities relative to their respective aliquots is shown in the plots in Figure 8 and Figure 9. It is clear that the Stanford projects have held an edge in system usage despite our efforts at resource allocation and the substantial voluntary efforts by the Stanford community to utilize non-prime hours. This reflects the development of the Stanford group of projects relative to those getting started on the national side and has correspondingly accounted for much of the progress in AI program development to date.

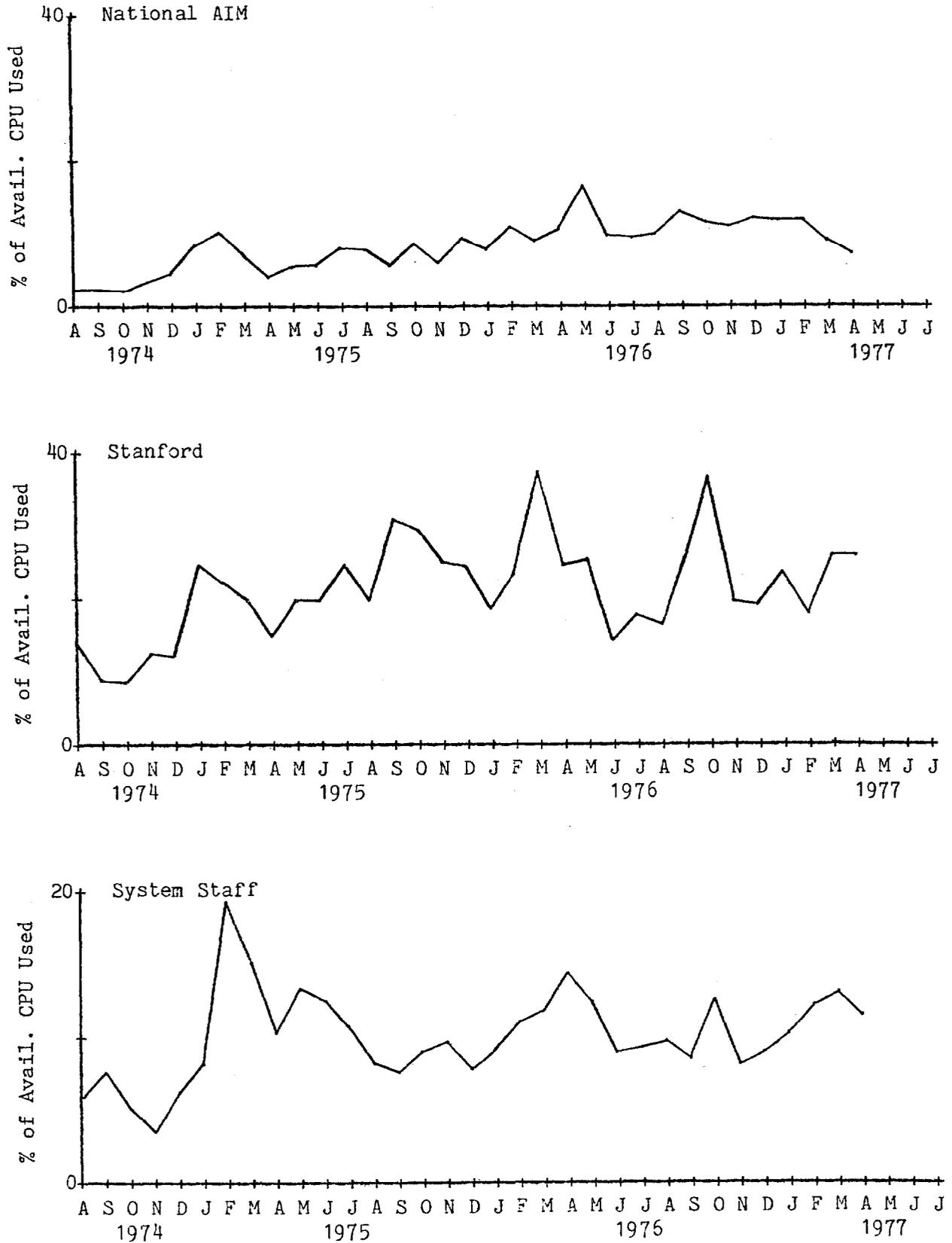


Figure 8. CPU Usage by Community

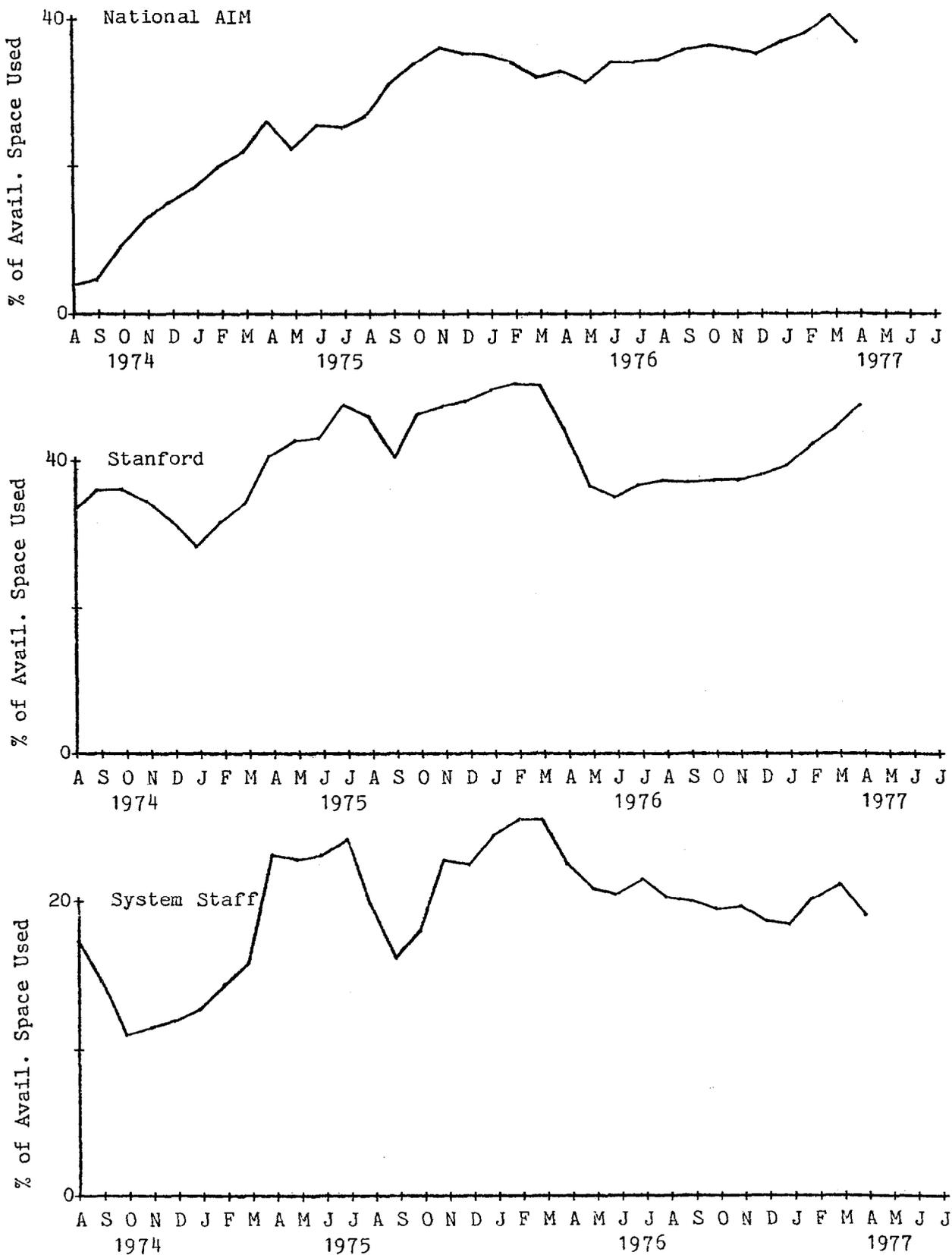


Figure 9. File Space Usage by Community

INDIVIDUAL PROJECT AND COMMUNITY USAGE

The table following shows cumulative 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, total CPU consumption by project (Hours), total terminal connect time by project (Hours), and average file space in use by project (Pages, 1 page = 512 computer words). These data were accumulated for each project for the months between May 1976 and April 1977. Again the well developed use of the resource by the Stanford community can be seen. It should be noted that the Stanford projects have voluntarily shifted a substantial part of their development work to non-prime time hours which is not shown in these cumulative data. It should also be noted that a significant part of the DENDRAL and MYCIN efforts, here charged to the Stanford aliquot, support development efforts dedicated to national community access to these systems. The actual demonstration and use of these programs by extramural users is charged to the national community in the "AIM USERS" category, however.

RESOURCE USE BY INDIVIDUAL PROJECT

STANFORD COMMUNITY	CPU (Hours)	CONNECT (Hours)	FILE SPACE (Pages)
1) DENDRAL PROJECT "Resource Related Research Computers and Chemistry" NIH RR-00612-08 (3 yrs. 1977-80) ARPA DAHC-15-73-C-0435 (2 yrs. 1977-79)	1181.64	19657.56	13058
2) HYDROID PROJECT "Distributed Processing and Problem Solving" ARPA DAHC-15-73-C-0435	40.92	924.49	239
3) MOLGEN PROJECT NSF MCS76-11649 NSF MCS76-11935 (2 yrs. 1976-78)	85.61	2487.73	1853
4) MYCIN PROJECT "Computer-based Consult. in Clin. Therapeutics" HEW HS-01544 (2 yrs. 1977-79) NSF (2 yrs. 1977-79)	410.87	5540.75	6688
5) PROTEIN STRUCT MODELING "Heuristic Comp. Applied to Prot. Crystallog." NSF DCR 74-23461 (2 yrs. 1977-79) ARPA DAHC 15-73-C-0435	159.80	2894.19	2477
6) AIHANDBOOK PROJECT	26.46	464.42	639
7) PILOT PROJECTS (see reports in Section 6.3 in Book II)	327.67	5919.33	3506
	-----	-----	-----
COMMUNITY TOTALS	2232.97	38988.47	28460

NATIONAL AIM COMMUNITY

1)	ACT PROJECT "Acquisition of Cognitive Procedures" NIMH MH29353 ONR N0014-77-6-0242	57.02	1195.84	986
2)	HIGHER MENTAL FUNCTIONS "Computer Models in Psychiatry and Psychother." NIH MH-27132-02 (2 yrs.) UCLA NPI Gen. Res.	206.03	2680.16	2198
3)	INTERNIST PROJECT (DIALOG) "Computer Model of Diagnostic Logic" BHRD MB-00144-03 (3 yrs.)	205.20	2721.25	3535
4)	MISL PROJECT "Medical Information Systems Laboratory" US-PHS-MB00114-03 (3 yrs.)	9.27	380.05	876
5)	RUTGERS PROJECT "Computers in Biomedicine" NIH RR-00643-05 (3 yrs.)	139.63	2433.43	10862
6)	SECS PROJECT "Chemical Synthesis"	308.96	4374.03	4515
7)	AIM PILOT PROJECTS (see reports in Section 6.4 in Book II)	40.91	1326.56	1558
8)	AIM Administration	11.13	383.22	1762
9)	AIM Users	56.89	672.35	352
		-----	-----	-----
	COMMUNITY TOTALS	1035.04	16166.90	26654

SUMEX STAFF AND SYSTEM

1) Staff	903.07	23198.86	11919
2) Miscellaneous	80.87	2508.98	1721
3) Operations	1505.50	63113.94	32382
	-----	-----	-----
COMMUNITY TOTALS	2489.44	88321.78	46022
	=====	=====	=====
RESOURCE TOTALS	5757.45	143977.15	101136

SYSTEM DIURNAL LOADING VARIATIONS

The following figures give a picture of the recent variations in diurnal SUMEX system load, taken during March 1977. The plots include:

- Figure 10 - Total number of jobs logged in to the system
- Figure 11 - Percent of total CPU time used by logged in jobs (maximum is 200% for dual processor capacity)
- Figure 12 - Percent of total CPU time consumed as overhead; I/O wait, core management, scheduling, etc. (maximum = 200%)
- Figure 13 - Balance set size (number of jobs in core)
- Figure 14 - Number of runnable jobs (whether or not in core)

The abscissa for these plots is broken into 20 minute intervals throughout the day. The ordinate for each interval is the average of all the daily measurements for that interval over the weekdays during March 1977. A daily measurement for a given 20 minute interval is in turn an average of the appropriate statistic sampled every 10 seconds. Since these plots display overall average data, they give representative illustration of the general characteristics of diurnal loading. There are, of course, substantial fluctuations in the quantities measured from day to day as well and for some, also on time scales shorter than the intervals displayed in the figures. For example in Figure 14, the number of runnable jobs (equivalent to the system "load average") shows a fairly smooth curve peaking at 6.7 jobs. On both a scale of minutes and from day to day, however, the number of runnable jobs will vary from only a few to 12 or more. This fluctuation is not shown in these average plots but also plays a role in the responsiveness of the system.

In the heading of each plot are shown range statistics for the measurement over various parts of the day. Range data include the minimum value "Low", average value "Ave", and maximum value "High". The first line of the heading gives the range over the whole day and on succeeding lines, "Prime Time" covers 6:00-18:00 Pacific time and "Non Prime Time" covers the remaining night time hours.

It can be noted in Figure 12 that the current overhead level for the dual processor system is quite high (about 33% per processor). This is because of the limited memory size (256K words) we currently have and the resulting increase in swapping interrupt rate and I/O wait time. We have a proposal pending with the AIM Executive committee to augment our memory which should reduce this overhead down to our earlier single processor levels (about 15-20% per processor).

Figure 10. Average Diurnal Loading (3/77): Total Number of Jobs

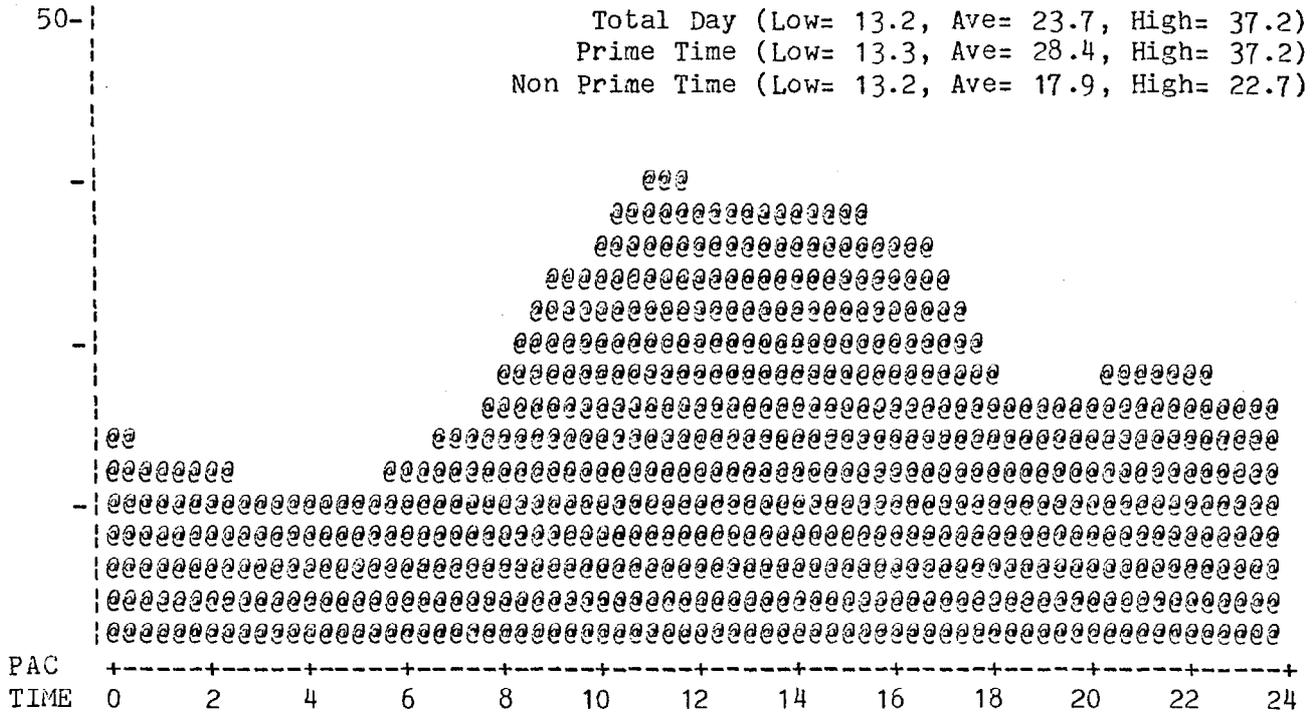


Figure 11. Average Diurnal Loading (3/77): Percent Time Used

