Figure   12.   Average Diurnal Loading (3/77): Percent Overhead
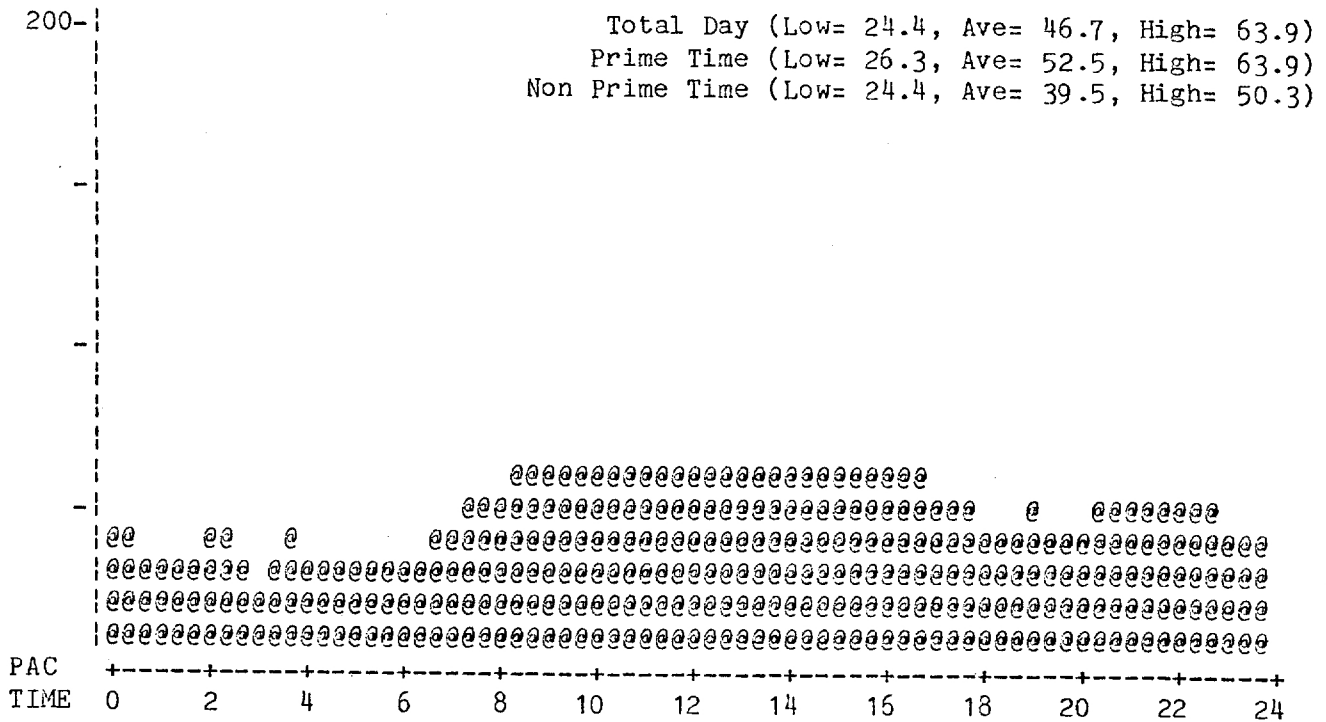
```
200-|
    |                          Total Day (Low= 24.4, Ave= 46.7, High= 63.9)
    |                          Prime Time (Low= 26.3, Ave= 52.5, High= 63.9)
    |                      Non Prime Time (Low= 24.4, Ave= 39.5, High= 50.3)
    |
  -|
    |
    |
    |
    |
  -|                       @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
  -|                     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@   @   @@@@@@@@
    |@@     @@    @         @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    |@@@@@@@@@  @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
PAC +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
TIME 0     2     4     6     8    10    12    14    16    18    20    22    24
```

Figure   13.   Average Diurnal Loading (3/77): Balance Set - Jobs in Core

```
12-|
   |                     Total Day (Low= .7, Ave= 2.4, High= 4.9)
   |                     Prime Time (Low= .7, Ave= 3.1, High= 4.9)
   |                 Non Prime Time (Low= .8, Ave= 1.6, High= 2.8)
   |
 -|
   |
   |
   |
   |
 -|
   |
   |                     @@@
   |                   @@@@@@@   @ @@@@@@
   |                  @@@@@@@@@@@@@@@@@@@@@
 -|                 @@@@@@@@@@@@@@@@@@@@@@@@@@@@@            @
   |               @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@      @@@@@@@
   |@             @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
   |@@@@@@@@   @@@@   @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
   |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
PAC +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
TIME 0     2     4     6     8    10    12    14    16    18    20    22    24
```
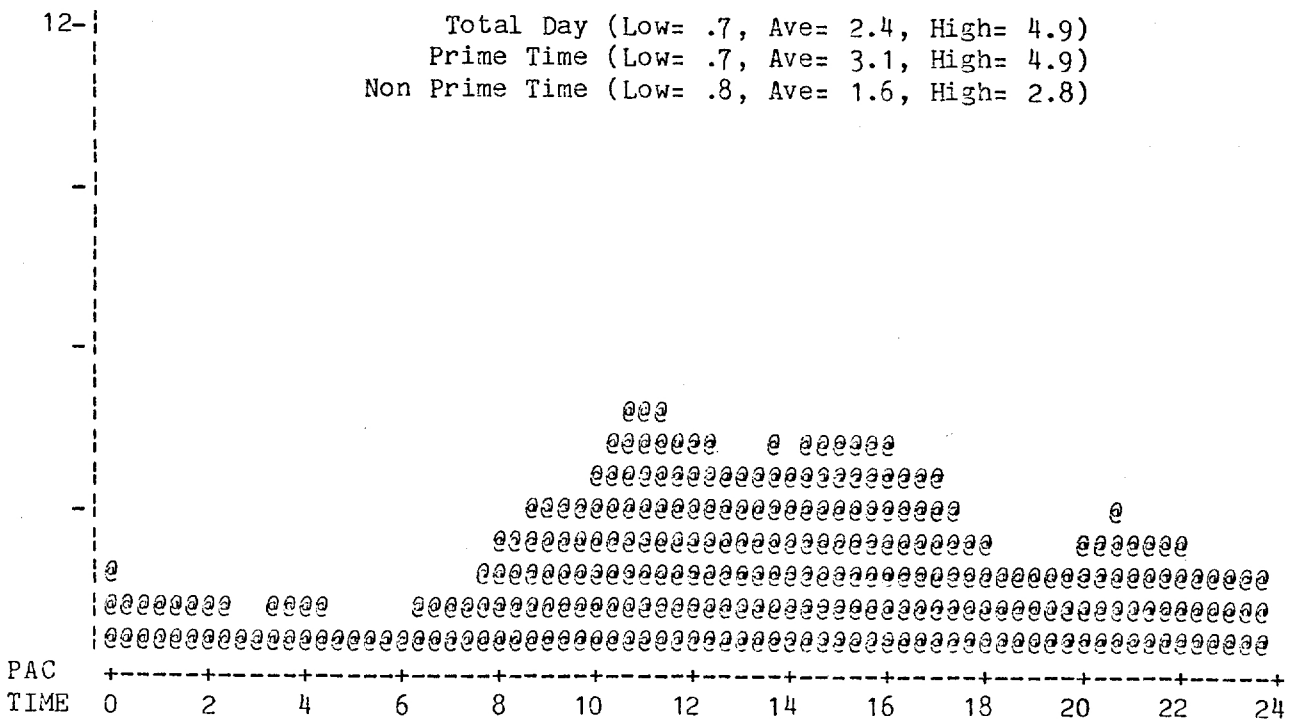
Figure   14.   Average Diurnal Loading (3/77): Runnable Jobs

```
12-|                        Total Day (Low= .7, Ave= 2.9, High= 6.7)
   |                       Prime Time (Low= .7, Ave= 3.8, High= 6.7)
   |                   Non Prime Time (Low= .8, Ave= 1.7, High= 3.1)
   |
   |
 -|
   |
   |
   |
 -|                              @
   |                            @@@@
   |                          @@@@@@@          @@ @
   |                         @@@@@@@@@@@@@@@@@@@@@@
   |                       @ @@@@@@@@@@@@@@@@@@@@@@@@
   |                        @@@@@@@@@@@@@@@@@@@@@@@@@@@@
 -|                        @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@            @@
   |                        @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@      @@@@@@@@@
   |@        @              @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
   |@@@@@@@@@ @@@@@          @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
   |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
PAC +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
TIME 0     2     4     6     8    10    12    14    16    18    20    22    24
```
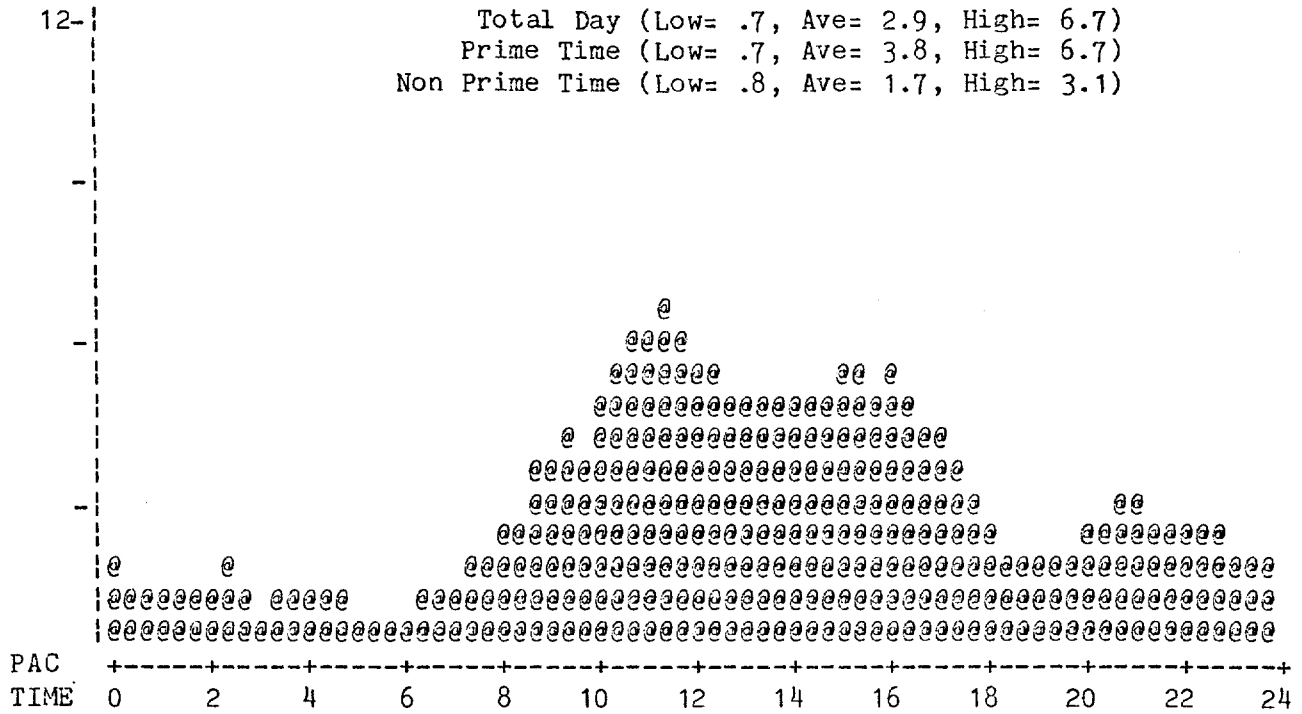
1.3.2.13   NETWORK USAGE STATISTICS

NETWORK USAGE PLOTS

The plots in Figure 15 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 1:12 over our period of usage. 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.
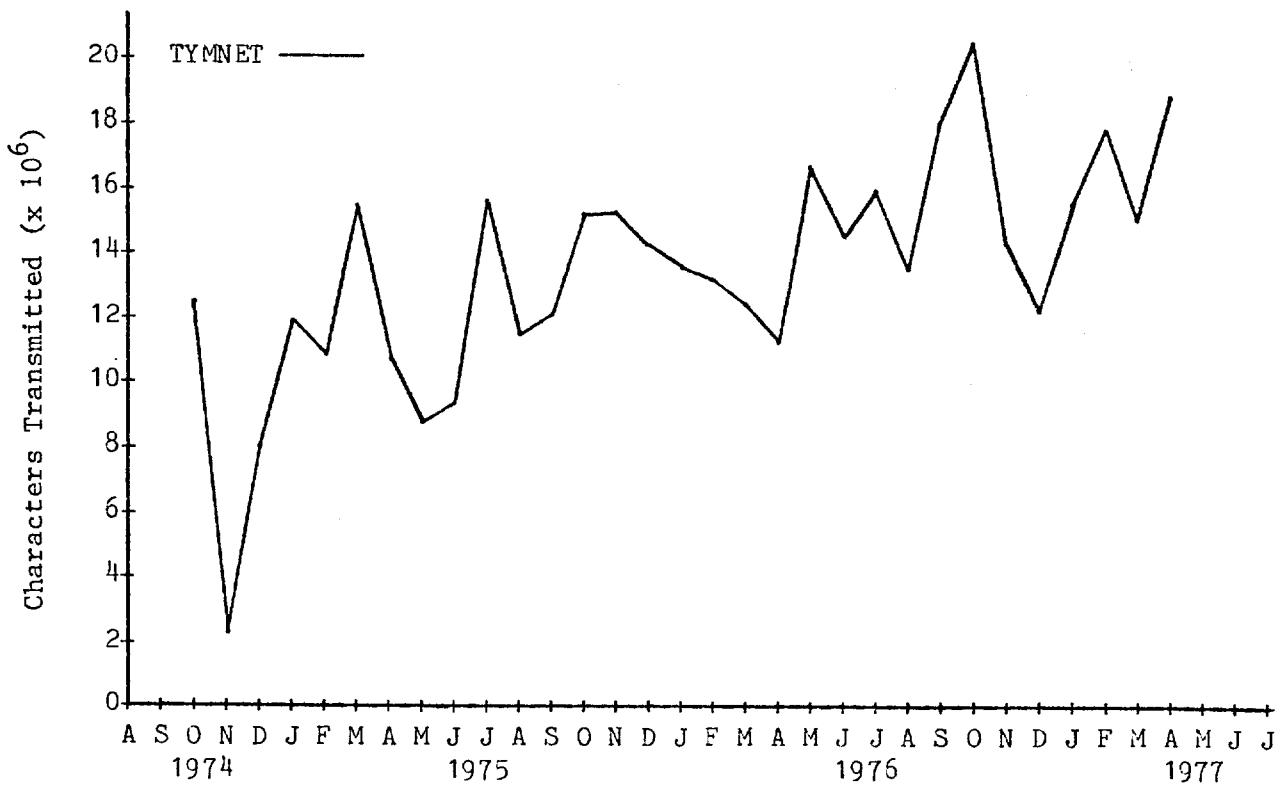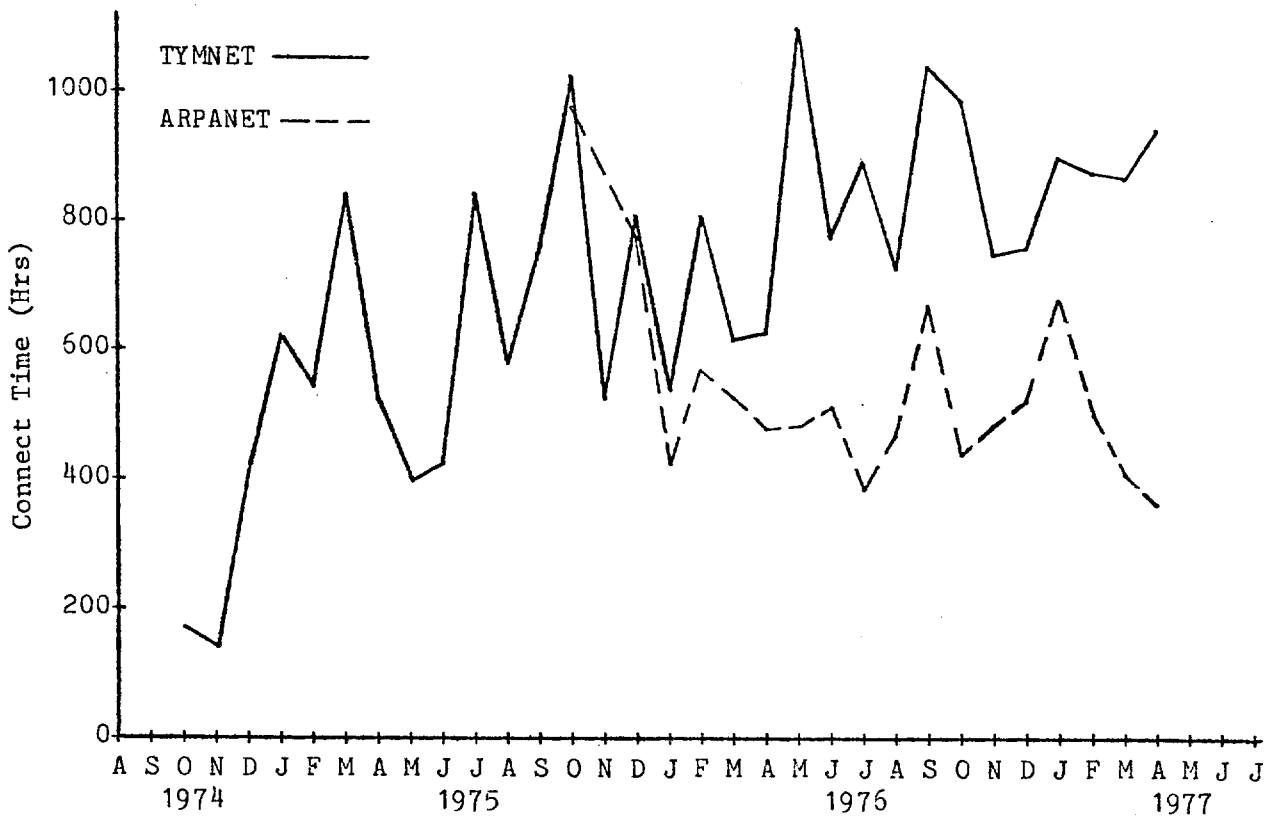
Figure 15.   TYMNET and ARPANET Usage Data

1.3.2.14   <u>PUBLICATIONS</u>

The following are publications for the SUMEX staff and have included papers describing the SUMEX-AIM resource and on-going research as well as documentation of system and program developments. Publications for individual collaborating projects are detailed in their respective reports (see Section 6 on page 41 in Book II).

[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

[3] Wilcox, C. R., "MAINSAIL - A Machine-Independent Programming System," Proceedings of the DEC Users Society, Vol 2, No 4, Spring 1976.

Mr. Clark Wilcox also chaired the session on "Languages for Portability" at the DECUS DECsystem10 Spring '76 Symposium.

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, the many subsystems available to users, and MAINSAIL. 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, and policy guidelines (see Appendix VI, and Appendix VII in Book II). Publications for individual user projects are summarized in the respective reports (see Section 6 in Book II).

1.3.2.15   RESOURCE STAFFING HISTORY

PROFESSIONAL PERSONNEL (YEARS 01-04)

Name	Title of Position	(*) % of Effort	Period of Appointment
Lederberg, Joshua	Principal Investigator	10	10/1/73 – present
Rindfleisch, Thomas	Facility Manager	100	10/1/73 – present
Levinthal, Elliott	AIM Liaison	22	12/1/73 – present
Cower, Richard	System Programmer	100	6/24/74 – 6/15/77
Crossland, James	System Programmer	100	8/6/74 – 1/16/76
Gilmurray, Frank	System Programmer	100	6/1/77 (tent. start)
Heathman, Michael	System Programmer	100	10/1/73 – 8/15/75
Lieb, James	System Programmer	100	7/1/74 – 11/14/75
Reiss, Steven	System Programmer	100	10/1/73 – 7/31/74
Sweer, Andrew	System Programmer	100	1/19/76 – present
Tucker, Robert	System Programmer	100	6/1/77 (tent. start)
Schulz, Rainer	System Programmer - IMSSS	61	2/1/74 – present
Roberts, Ronald	System Programmer - IMSSS	50	2/1/74 – 7/31/74
" "	" " "	52	5/1/75 – 7/31/75
Smith, Robert	System Programmer - IMSSS	50	5/1/75 – 7/31/75
Quam, Lynn	Syst. Prog. - Cardiology	50	3/1/76 – 5/31/76
Johnson, Suzanne	Applications Programmer	100	7/22/74 – present
Smith, Nancy	Applications Programmer	100	3/25/74 – 8/20/76
Kahler, Richard	User Consultant	100	12/1/75 – present
Jackson, Phillip	User Support Specialist	100	11/18/74 – 7/28/75
Wilcox, Clark	Syst. Prog. - Res. Asst.	63	3/25/74 – present
Veizades, Nicholas	Electronics Engineer - IRL	50	10/1/73 – present
Nozaki, Thomas	Electronics Engineer - IRL	6	5/1/74 – present

(*) The figures shown give the % of effort during the respective periods of
    employment.

2 SPECIFIC AIMS

The following outlines the specific objectives of the SUMEX-AIM resource during the follow-on five year period. Note that these objectives cover only the resource nucleus; objectives for individual collaborating projects are discussed in their respective reports (see Section 6 on page 41 in Book II). We break our research aims into the categories 1) resource operations, 2) training and education, and 3) core research.

2.1 RESOURCE OPERATIONS AIMS

The broad objectives remain to provide an effective computing facility with extensive network access to support the community of projects developing AI applications in medicine. This goal includes the limited dissemination of these programs to outside research groups to provide the necessary feedback from actual research applications for effective program development. Specific aims include:

1) Continue the building of a community of projects applying AI techniques to medical problems including improving mechanisms for inter- and intra-group collaborations and communications. We plan to extend the existing AIM community management structure to accommodate justified growth in computing resources at other sites including a close collaboration between nodes on such a "resource network" and a meaningful division of responsibilities and regional expertise. To minimize administrative barriers to the community-oriented goals of SUMEX-AIM, we plan to retain the current user funding arrangements; user projects will fund their own manpower and local needs and will actively contribute their special expertise to the SUMEX-AIM community in return for an allocation of computing resources under the control of the AIM management committee structure. There will be no "fee for service" charges for community members. While AI is our defining theme, we may entertain exceptional applications justified by some other unique feature of SUMEX-AIM essential for important biomedical research.

2) Provide an effective computing resource to support the development and research dissemination of large and complex computer programs for a broad range of medical AI applications. This will include the continued development and refinement of the existing resource and the development and implementation of a plan for the upgrade of current hardware to the emerging next generation when justified by community, technical, and economic advantages.

3) Provide effective and geographically accessible network communication facilities to the SUMEX-AIM community for effective remote collaborations and to allow external users to experiment with available AI programs. We also plan to demonstrate the utility of network communications for scientific collaboration, in selected cases which do not interfere with our primary mission, to groups in other areas of computer science related to medicine. The ONET collaboration (see the Rutgers Resource progress

report on page 144) illustrates the value of these facilities apart from the AI programs themselves.

2.2   TRAINING AND EDUCATION AIMS

Our goals during the follow-on period for assisting new and established users of the SUMEX-AIM resource are a continuation of those adopted for the first grant term. Collaborating projects will provide their own manpower and expertise for the development and dissemination of their AI programs. The SUMEX resource will provide community-wide support and will work to make resource goals and AI performance programs known and available to appropriate medical scientists. Specific aims include:

1) Provide documentation and assistance in interfacing users to resource facilities and programs. We will continue to exploit particular areas of expertise within the community for developing pilot efforts in new application areas.

2) Continue to allocate "collaborative linkage" funds to qualifying new and pilot projects to provide for communications and terminal support pending formal approval and funding of their projects. These funds are allocated in cooperation with the AIM Executive Committee reviews of prospective user projects.

3) Provide support for a "visiting scientist" position to allow prospective qualified SUMEX-AIM project investigators or users to spend a term in close contact with on-going research work. The selection of appropriate candidates for this rotating position would be made in cooperation with the AIM Executive Committee.

4) Continue to support AIM Workshop activities in collaboration with the Rutgers Computers in Biomedicine resource.

2.3   CORE RESEARCH AIMS

Our core research efforts will emphasize the generalization and documentation of tools and techniques available for AI research and applications and the examination of alternative approaches for implementing and exporting large and complex AI performance programs. These efforts will be important community-wide to facilitate the investigation of new application areas and to meet the demand, beyond SUMEX-AIM capacity, for external users to be able to run developed AI programs conveniently. Fortunately, we have independent funding from various agencies for research activities that overlap the core-research

opportunity, e.g., CONGEN, MOLGEN, Heuristic Programming Project, and DENDRAL mass spectrometry. Specific aims include:

1) Continue to <u>encourage</u> <u>community</u> <u>efforts</u> <u>at</u> <u>organizing</u> <u>and</u> <u>developing</u> <u>AI</u> <u>techniques</u> by supporting projects such as the AI Handbook, special language developments (e.g., KRL), and other projects community members may propose to contribute.

2) Explore the <u>generalizations</u> <u>of</u> <u>AI</u> <u>tools</u> for knowledge acquisition, representation, and utilization; reasoning in the presence of uncertainty; strategy planning; and explanations of reasoning pathways. This effort will attempt to extract and generalize some of the best concepts and functional capabilities developed in the context of particular projects (e.g., DENDRAL, MYCIN, MOLGEN, etc.). The objective is to evolve a body of software packages that can be used to more efficaciously build future knowledge-based systems and explore other medical AI applications.

3) <u>Explore</u> <u>AI</u> <u>software</u> <u>implementation</u> <u>and</u> <u>export</u> <u>mechanisms</u> such as network communication systems, machine-independent languages, and special purpose computer systems. This will include the continued development of the MAINSAIL system and the investigation of microprogrammable machines specialized for target languages or satellite general purpose machines capable of running existing systems. Even the present level of computer capacity is not sufficient to meet the demands of a number of our projects. The DENDRAL CONGEN program is a good example where the potential for effective application to real biochemical structure determination problems is close but it simply takes too long to run problems that are really interesting. Therefore new approaches to computing are needed that may involve parallel processing, multiple small machines, or new developments from commercial vendors such as very much cheaper analogs of the PDP-10 that could be run in a more nearly dedicated mode.

3   METHODS OF PROCEDURE

This section details our plans for SUMEX-AIM goals during the next five year period. As indicated earlier, objectives and plans for individual collaborating projects are discussed in Section 6 on page 41 (see Book II). In general SUMEX-AIM will retain its community orientation in formulating and implementing a resource for AI research in medicine. We have had good success at integrating the tools and expertise of on-going active research efforts where possible and building on these where extensions or innovations are necessary. This orientation has proved to be an effective way to build the current facility and community and we expect it to be equally productive during the next period. We have assembled a growing community of projects which contribute to SUMEX-AIM resource goals and have at the same time come to depend on SUMEX for computing support and as a means of interacting with collaborators. We plan to continue our commitment to providing effective support to this community of projects.

This opportunistic approach also places constraints in synchronizing particular advances with our community needs. We are presently facing demands for increased computing resources as well as for effective methods for exporting mature AI performance programs. At the same time a new generation of hardware and firmware systems is just becoming available. These will have a large impact as a means to meet our goals, providing economic and technical advantages while minimizing redesign and reprogramming requirements. The anticipated timing for the announcement of a new generation of general purpose machines that might run AI software using existing operating systems and language support with substantially reduced capital investment is one to two years off. Such systems could be used to export software packages intact or to incrementally augment central resources like SUMEX. A similar situation exists for special purpose microprogrammable machines which can be tailored to particular language needs for increased throughput and efficiency. We aim to respond in a timely fashion to take advantage of this emerging technology but until concrete details are publically available, we can only describe our basic objectives and general design possibilities.

Thus the following description of research plans concentrates on software issues in planning for assimilation of the new technologies with the expectation that hardware announcements one to two years hence will impel careful reconsiderations of our strategies. Detailed budgets for computing hardware conversions are only approximate pending more detailed information on pricing. Our approach is to describe the research concept and gross estimated funding required, for review of these objectives at this time. We will further refine and elaborate the details of these plans during the first one to two years of the grant and submit them through the AIM Executive and Advisory Committees and the NIH Biotechnology Resources Program Office for approval prior to implementation.

3.1   RESOURCE OPERATIONS PLANS

3.1.1   SYSTEM HARDWARE AND MONITOR PLANS

As discussed in the progress section and supported by collaborating project reports, we have implemented an effective computing resource to support AI applications to medical research. We have augmented the present system to increase its effective capacity as far as we economically can to meet community needs. We do not propose any substantial changes either in scope of the existing resource or in its capacity. Other members of our community have proposals pending for other regional centers which may be justified on their own merits and the needs of the AIM community. We support the development of such regional expertise and specialization where justified which may allow a more coherent adaptation of a particular facility's resources to the needs of a subset of the AIM community. For example, a substantial group of biochemical structure analysis projects has grown up (DENDRAL, Chemical Synthesis Project, Protein Structure Project, and Molecular Genetics Project) as well as a group of medical diagnostic projects (MYCIN, Rutgers ONET, and INTERNIST as well as several pilot efforts). If regionalization becomes indicated, AIM facilities could be reoriented to serve the special needs of these research and target communities via separate systems, while maintaining close administrative and informational ties. We cannot predict the funding support such new facilities might receive but we will cooperate fully in getting them started and in assuring effective management for the benefit of the overall AIM community.

Our own facility has operated at capacity since early in our present grant term owing to the continuing maturing of on-going projects and the recruitment of new users, despite the periodic augmentation. As indicated earlier, our present hardware cannot be augmented further without upgrades to major mainframe and memory components. This should be done only after optimizing with respect to available new systems which are scheduled for announcement in the next year or so. There have been a number of recent relevant announcements but these machines have not yet been of a capacity or economic advantage to warrant immediate upgrade (indeed our decision to develop the dual KI-10 processor system was made on the basis of optimum cost-effectiveness within current technology and budgets). Furthermore, these systems are being sold packaged with relatively expensive memory and file storage and future releases may allow a more cost-effective mix of components from multiple vendors.

Our hardware design is now approximately five to six years old and will be twelve years old by the end of the follow-on 5 year grant term. The economics and technical performance of the newer systems, the evolving software gaps from inherent backward incompatibilities, and the reliability and maintainability of our existing equipment will pose new opportunities and problems. They may point to a strong rationale for an upgrade of the SUMEX-AIM system to meet the needs of the AI community we are supporting. The costs of this new generation of hardware will represent a progressively smaller part of the overall effort, compared to human resource inputs, especially if user participation is fairly weighted.

The TOPS-20 system DEC is currently marketing is derived from TENEX but already, DEC has made changes which cause incompatibilites with earlier systems. Many of these are in the direction of improved system performance (file system redundancy, system call enhancements, etc.) while others are of less obvious value (file naming conventions, message file formats, etc.). Whatever the reason, DEC's TOPS-20 system will likely dominate future system purchases and will increasingly diverge from ours. This causes a larger burden in our pursuit of software sharing and will affect the ease with which we can cooperate with other potential AIM network nodes. To avoid effective isolation, we will have to maintain effective compatibility. DEC has no plans for making TOPS-20 run on KI-10's and it is not likely others will undertake this within the currently strict licensing restrictions and DEC's motivations to sell KL-10's. Our apparent alternatives are to upgrade to some KL-"n" system when this product line matures and fills out so a proper choice can be made or to progressively modify our current system to remain as compatible as possible. A hardware conversion would likely cost at least $500,000 (based on current prices, but presumably much less as time passes) while system modifications for compatibility will entail 1-2 additional people per year in software effort. The cost of the latter approach must also include a measure of user community investment to circumvent unavoidable residual incompatibilities. The choice for optimum return will depend on the timing of major price declines for a given hardware capability, and on the way that cognate facilities evolve and participate in sharing software burdens.

We do not expect these trade-offs to be clear before 1979. We tentatively propose to expend the man-effort required to maintain compatibility between our existing system and TOPS-20 so long as this remains tenable. We budget initially one person for this purpose and add an additional programmer at the middle of the grant term. If this approach proves too costly and ineffective, we may propose reallocating these funds for a hardware conversion. Such a contingency would be thoroughly reviewed with AIM management committees and the NIH-BRP before finalizing a plan or requesting additional funding.

In the meantime we plan to reevaluate the performance of our existing system to wring out any remaining inefficiencies for more effective community support. The dual processor system has stabilized nicely and with the memory augmentation we are implementing, we will have taken advantage of all of the obvious sources of inefficiency. We will rereview the detailed operation of the facility to try to uncover remaining areas of cleanup. Recent measurements show that a high percentage of available time (80-90% in one recent test) is spent in various system routines which provide the rich set of monitor calls available through the TENEX system. It is therefore important to optimize the efficiency of the most widely used calls.

We also plan as part of this investigation to examine alternative strategies for managing memory allocations to running jobs. This will include attempting to minimize paging overhead by preloading job working sets to better utilize and overlap swapping I/O with other activities rather than waiting for page faults to read in pages on demand. We will also consider giving some program control over working set definition.

3.1.2   COMMUNICATION NETWORK PLANS

Networks remain centrally important to the research goals of SUMEX-AIM. We have had good success at meeting the geographical needs of the community during the early phases through our ARPANET and TYMNET connections. The major problems focus on terminal interaction delays through relatively slow or congested network facilities. In the next year or so TYMNET will be announcing their upgraded network (TYMNET II) which may offer additional advantages for our community such as higher terminal speeds, more dynamic terminal routing, and inter-host communications. If additional AIM servers are implemented, it will be important to coordinate their network access with that of SUMEX for effective user interactions and system collaborations.

During this same period ARPANET may be undergoing similar redesigns and possible further specialization to defense needs. In parallel, the TELENET facilities are evolving rapidly and whereas they offer a symmetric service for file transfer and terminal traffic, character delays are currently too high to warrant connecting immediately. We expect to retain our present connections over the early phases of the follow-on grant and to evaluate new upgrades as they become available. The specific goals for this upgrade will be improved terminal support and effective file transfer mechanisms available community-wide, particularly to interact with other AIM nodes.

3.1.3   SOFTWARE SUPPORT PLANS

We will continue to maintain the system, language, and utility support software on our system at the most current release levels, including up-to-date documentation. We will also be extending the facilities available to users where appropriate, drawing upon other community developments where possible. We rely heavily on the needs of the user community to direct system software development efforts. Two specific areas we plan to pursue are extensions to the bulletin board system and improved facilities for managing and organizing collections of related information as for example, program libraries and documentation, bulletin board or message files, collections of user profile information, etc. Bulletin board extensions will include improved facilities for searching for relevant information, associating a given bulletin with multiple topic labels, and more effectively apprising users of new information of interest. We are also examining extensions of the TENEX file system syntax and design to allow better logical organization and access to groups of file information. This may include facilities to define a hierarchical data structure, a"file system within a file", to name and manipulate logically related but independent pieces of information. A number of programs use ad hoc directories to access segments of information. We would hope to better standardize and improve such tools.

3.1.4   COMMUNITY MANAGEMENT PLANS

We plan to retain the current management structure that has worked out well
for the recruitment and review of new projects and the guiding of resource policy
formation. We expect the Executive and Advisory Committees to play a continuing
important role in advising on priorities for facility evolution and on-going
community development efforts such as MAINSAIL in addition to their recruitment
efforts. The composition of the Executive committee will grow as needed to
assure representation of major user groups and medical and computer science
applications areas. The Advisory Group membership rotates with each member
serving one to two years and spans both medical and computer science research
expertise. We expect to maintain this policy.

The AIM workshops under the Rutgers resource have served a valuable
function in bringing community members and prospective users together. We will
continue to support this effort in terms of the Stanford community participation
and providing a computing base for workshop demonstrations and communications.

3.2   TRAINING AND EDUCATION PLANS

We have an on-going commitment, within the constraints of our staff size, to maintain a high level of documentation of the evolving software support on the SUMEX-AIM system and to provide user help facilities such as the HELP and Bulletin Board systems. These latter aids are the best way we can assist resource users to find the information they need when they need it to solve access problems. Since much of our community is geographically remote from our machine, these on-line aids are indispensible for self help. We will also provide on-line personal assistance to users within the capacity of available staff through the SNDMSG and LINK facilities.

We allocate funds in our budget to continue the "collaborative linkage" support initiated during the first term of the SUMEX-AIM grant. These funds are allocated under Executive Committee authorization for terminal and communications support to help get new users and pilot projects started.

We also have requested support for a "visiting scientist" position which will allow selected prospective investigators to gain first hand experience by visiting on-going projects such as at Stanford. We feel this can serve an important role in catalyzing the development of new application areas and in disseminating the AI programs and techniques developed within the SUMEX-AIM community. The selection of appropriate individuals will be coordinated with the AIM committees as well.

Finally, we will continue to actively support the AIM workshop series in terms of planning assistance, participation in program presentations and discussions, and providing a computing base for AI program demonstrations and experimentation.

3.3   CORE RESEARCH PLANS

3.3.1   GENERALIZATION OF AI TECHNIQUES

The SUMEX-AIM facilities have made it possible to explore many of the
frontiers of Artificial Intelligence research within the context of specific
systems of medical relevance. Among those issues are the acquisition,
representation and utilization of knowledge (both formal and judgmental),
reasoning under uncertainty, explanation of a program's reasoning steps, and
strategy planning. During the next period we wish to extract some of the best
concepts and programming techniques from the specific programming systems,
demonstrate their generality by incorporating them into other working programs,
and design and implement packages which can be used to construct other high
performance, knowledge based systems.

The five projects described below are proposed as basic core research in
support of the various AIM community projects applying the techniques of AI
research to biomedical problems. References for this material can be found on
page 76. Because these projects are extensions of on-going work, we are able to
generalize from existing programs without requesting support for maintenance or
development of the programs themselves. This is another example of the
synergistic community interactions of the SUMEX-AIM resource.

3.3.1.1   DESIGN OF KNOWLEDGE-BASED CONSULTATION SYSTEMS

Objective

Recent work has suggested that one key to the creation of intelligent
systems is the incorporation in programs of large amounts of task-specific
knowledge. We intend to develop (i) methods of using large stores of expert
knowledge as a foundation for computer-based reasoning, and (ii) methods of
facilitating the knowledge transfer from human experts to computer programs. We
believe that this will lead to principles that may help turn the art of building
large systems into more of a science, and thus aid other investigators who are
building large knowledge-based systems. To do this, we will work on a number of
problems involving knowledge representation, accumulation, management, and use,
in the context of a software "laboratory" designed to facilitate the construction
and use of large knowledge bases.

Motivation

Some of the earliest work in artificial intelligence centered around the
attempts to create generalized problem solvers. Work on programs like GPS
[Newell72] and theorem proving [Nilsson71], for instance, was inspired by the
apparent generality of human intelligence and motivated by the belief that it
might prove possible to develop a single program applicable to all (or most)
problems. While this early work demonstrated that there was a large body of

useful general purpose techniques (such as problem decomposition into subgoals, and heuristic search in its many forms), these techniques did not by themselves offer sufficient power for high performance.

Recent work has instead focussed on the incorporation of large amounts of task specific knowledge in what have been called "knowledge-based" systems. Rather than non-specific problem solving power, knowledge based systems have emphasized high performance based on the accumulation of large amounts of knowledge about a single domain.

A second successful focus in work on intelligent systems has been the emphasis on the utility of solving "real world" problems, rather than artificial problems fabricated in simplified domains. This is motivated by the belief that artificial problems may prove in the long run to be more a diversion than a foundation for further work, and by the belief that the field has developed sufficiently to provide techniques that can aid working scientists. While artificial problems may serve to isolate and illustrate selected aspects of a task, solutions developed for those selected aspects often do not generalize well to the complete problem.

There are numerous current examples of successful systems embodying both of these trends, systems which apply task-specific knowledge to real world problems. They include efforts at symbolic manipulation of algebraic expressions [Macsyma74], speech understanding [Lesser74], chemical inference [Buchanan71], and interactive consultants in a few specific areas [Pople75, Shortliffe76].

While all of these systems display an encouraging level of performance, however, two fundamental problems remain. First, assembling the knowledge base for each of these is a difficult, continuous task that has in most cases extended over several years. Second, the result of this effort is typically a system with an impressive level of performance, but only within a sharply limited domain of application. High performance has been achieved at the cost of generality and man-years of work in knowledge base construction.

But if programs require large stores of knowledge for high performance, can we take a step back and discover powerful and broadly applicable techniques for accomplishing this transfer of knowledge? That is, can we discover ways of facilitating the communication, management and use of large amounts of task-specific knowledge? The result would be an intelligent system whose generality arose from access to the appropriate human experts, and whose power was based on the store of knowledge it acquired from them.

Two central themes of the proposed work are facilitating knowledge base construction and improving the generality of the reasoning programs that use the knowledge base. We intend to employ a computer system based on broadly applicable techniques for knowledge encoding and use, and couple it with powerful techniques for accomplishing the transfer of knowledge from human experts to computer programs. The foundation for the computer system will be provided by the domain independent core of the Mycin system [Shortliffe76, Davis77]. This will be the basis for a software "laboratory" in which we can examine the relevant issues of knowledge representation, accumulation, management, and use. By setting this work in the context of a specific, existing body of software, a number of a very general issues become focussed into specific questions. Since

the program that constitutes our "laboratory" has been demonstrated to have a
strong degree of domain independence, the results of this work will be widely
applicable.

This should produce a new form of generality. Unlike GPS, we do not offer
one program which can solve problems in any domain. Rather, we offer the
foundation for a system, along with a methodology for instantiating that system
in any one specific domain. The foundation and methodology provide a framework
for the expression, management, and use of domain specific knowledge, to make
this instantiation task a reasonable one. It is there in the foundation and the
methodology that our generality lies, not in the final performance program which
results.

3.3.1.2   ATTEMPT TO GENERALIZE (AGE) PACKAGE

The objective of this research is to isolate inference, control and
representation techniques from previous knowledge-based programs; reprogram them
for domain independence; write a rule-based interface that will help a user
understand what the package offers and how to use the modules; and make the
package available to SUMEX users, other research groups engaged in knowledge-
based systems development, and the general scientific community.

Detailed Discussion:

The goal of this new effort is to construct a computer program to
facilitate the building of knowledge-based systems. The design and
implementation of the program will be based primarily on the experience gained in
building knowledge-based systems at the Heuristic Programming Project in the last
decade. The programs that have been built are: DENDRAL[Buchanan71], meta-
DENDRAL[Buchanan72], MYCIN[Shortliffe76], AM[Lenat76], HASP[Nii77], Protein
Structure Modeler[Engelmore77], and MOLGEN[Stefik77] (the latter two currently
under development). Initially, The AGE program will embody methods used in our
programs. However, the long-range objective is to integrate methods and
techniques developed at other A.I. laboratories. The final product is to be a
collection of useful "building-block" subprograms, combined with a knowledge-
based front-end that will assist a user in constructing knowledge-based programs.
It is hoped that AGE can speed up this process and facilitate transfer of the
technology by: (1) packaging common AI software tools so that they do not need to
be reprogrammed for every problem; and (2) helping people who are not knowledge-
engineering specialists to write knowledge-based programs.

Two Specific Research Activities of the AGE Effort are:

1. The isolation of techniques used in knowledge-based systems. It has always
    been difficult to determine if a particular problem-solving method used in
    a knowledge-based program is "special" to a particular domain or whether
    it generalizes easily to other domains. In the currently existing
    knowledge-based programs the domain-specific knowledge and the
    manipulation of such knowledge using AI techniques are often so closely

coupled that it is difficult to make use of the programs for other
domains. We need to isolate the AI techniques that are general to
determine precisely the conditions for their use.

2. Guiding users in the application of these techniques. Once the various
techniques are isolated and programmed for use, an "intelligent front end"
is needed to guide users in their application. Initially, we assume that
the user understands AI techniques and knows what he wants to do, but that
he does not understand how to use the AGE program to accomplish his task.
The program at this stage of the development will need to have the basic
tools coupled with a package to guide the user in applying these tools. A
longer-range interest involves helping the user determine what techniques
are applicable to his task. That is, we assume that the user does not
understand the necessary techniques of writing knowledge-based programs.
Some questions to be posed are: What are the criteria for determining if a
particular application is suited to a particular problem-solving
framework? How do you decide the best way to represent knowledge for a
given problem?

There are some smaller, but by no means trivial, questions which also need
answering. Is there a "best way" to write production rules which would
apply to many task domains? Is there a data representation that would
cover many tasks? What is the best way to handle differences in the
ability of the users of the AGE program?

Research Plan:

    The AGE program will be developed along two separate fronts, both of which
are divided into incremental development stages. The first of these fronts is
the development of the ability to help build many different types of knowledge-
based programs (the "generality" front). The second front is the development of
"intelligence" in the interaction between the user and the AGE program; i.e.
moving from dialogues on "how to use the tools in AGE" to "what tools to use"
(the "how-to-what" dialogue front). The proposed development plan contains the
following stages:

a. Generality: The development of a program package that will enable the user
    to build "HASP-like" knowledge-based programs characterized by the
    integration of multiple sources of knowledge, multi-level representation
    of solution hypotheses, opportunistic problem-solving methods, and
    explanation capability of the reasoning steps. The HASP-like paradigm has
    been used to solve problems of interpreting large amounts of digitized
    physical signals, but can also be extended to problems of processing large
    amounts of symbolic data.

    Dialogue: The development of dialogue to show the user how to utilize the
    packaged components in AGE to build HASP-like programs. The interactive
    capability will be limited to: specifying how to build multi-level
    hypothesis structure; how to write production rules to represent domain
    knowledge; and how to use various techniques available for opportunistic
    hypothesis formation.

b. Generality: Supplement the ability to build HASP-like programs with a
capability to build MYCIN-like goal oriented programs.

Dialogue: Same level of dialogue capability with additional ability to
discuss how to chain rules and how to specify the necessary parameters for
the context tree.

c. Generality: Same level as for b., i.e. ability to build HASP-like, MYCIN-
like or combination of HASP- and MYCIN-like knowledge-based programs.

Dialogue: Begin to extract from the user some key characteristics of the
task, and using that information begin to suggest appropriate knowledge
representation and problem-solving techniques for the user's task. This
interactive capability will be limited to the generality level at this
point in the AGE development.

d. Test phase: Test the usefulness of the AGE system by developing an
application program in some task domain. (a) An application program will
be chosen from among on-going program development efforts within our own
project or within the SUMEX-AIM community. An application will be chosen
whose primary task is that of interpreting large amounts of symbolic data
or described signal data. (b) Collect specific knowledge needed for the
application program and begin to develop the program using the AGE system.

3.3.1.3   PLAN PACKAGE

The PLAN package is oriented toward the representation of plans-of-action
and toward an expert's knowledge of the best problem solving strategies to employ
in his domain. A feature of the package is its ability to make inferences on
components of planning and strategy rules so that new plans and strategies can be
constructed readily from previous ones. The representation will allow the
manipulation of various "levels of detail" of plans and strategies. The package
will be made available as previously mentioned in connection with AGE.

Detailed Discussion:

Before starting a technical presentation of the ideas for the Plan Package,
it is worth highlighting some of the issues which motivate its development.

a. How can a variety of types of domain actions be accommodated in a
knowledge base?

b. How can a variety of types of strategy and control knowledge be
incorporated in a knowledge base?

c. How can a variety of types of problem solving states be expressed and
manipulated by the system?

d. How should plans be represented?

e. How can the problem statements for a variety of types of problems be acquired?

f. How does the expression and representation of problem solving states relate to the expression of the domain and strategy knowledge?

The Plan Package consists of two major entities -- the Planning Network and the Strategy Package. The Planning Network is a set of software which manages the representation of the plans created during the problem solving process. When a problem is acquired from a user, it is represented as an initial planning network. Problem solving takes place as the active strategy rules manipulate the planning network to create solutions. The Strategy Package itself is discussed in the next section.

Since the planning state knowledge is important for the expression of strategy in the Plan Package, it is worthwhile exploring briefly the nature of this knowledge. It is useful to consider the planning network as being composed of three parallel planes -- the solution plane, the planning plane, and the focus plane. These planes contain (1) the solution steps (domain rule applications) and world states, (2) the planning and design steps and (3) the focus of attention knowledge respectively. All three planes of the network are built dynamically during the problem solving process. Different types of nodes in the network correspond to the different components of the problem solving process.

A number of issues have been raised about the management of strategy knowledge.

a. How should strategies be expressed?

b. How can strategy information be assimilated so that the system will use it appropriately when designing or explaining solutions?

c. How can a knowledge based system assist a domain expert in structuring and expressing his ideas about strategy?

Means-ends analysis is one of the simplest ideas in the current stock of methods for problem solving. As such, it should exist as a standard strategy in a strategy package of artificial intelligence techniques to be used as needed. The current state of artificial intelligence, where a researcher must re-code Means-ends analysis any time he wishes to use it is akin to a carpenter forging a new hammer for each job.

One approach for making an instance of Means-ends analysis available as a tool would be to provide a packaged program which accepts arguments for the various components of Means-ends analysis (e.g. a difference table, difference function, etc.). The alternative being proposed here is a system which uses schemata to drive the strategy acquisition process and which can guide a user through the details. The goal is to create a supportive environment for the painless testing of fairly high level strategies. Such a system should be able to draw on its knowledge base to provide assistance in casting a problem into a Means-ends framework.

In summary, other systems have stumbled over the expression of more complex forms of domain and strategy rules and have been limited to solving a single kind of problem. We propose extending this work by developing what we have termed the Plan Package. The Plan Package consists of two major components - a schema-based representation for the problem-solving states termed the Planning Network and a schema-based representation for domain rules and strategies termed the Strategy Package. The Planning Network will provide a representation for a variety of types of problem solving so that the problem solving system will be able to solve more than one type of problem. The Strategy Package will provide a set of standard artificial intelligence strategies in the form of schemata, which may be instantiated into strategy rules when they are supplied with the particulars of domain knowledge. These schemata will facilitate the acquisition of tailored strategies by guiding a user a step at a time through the particulars of the acquisition process.

The Plan Package will be developed and tested in the domain of molecular genetics as part of the MOLGEN project. It will be further developed and extended to other domains as a test for generality as part of the AGE project.

3.3.1.4   HEURISTIC KNOWLEDGE ACQUISITION

Automatic Rule Formation Methods

Given a body of data from which rules are to be formed, together with a basic approach to rule induction, there remains a range of ways in which the data may be utilized, which differ in the degree of parallelism involved in the examination of instances. At one extreme are methods in which rules are formed and refined in a sequence of steps, each step involving the examination of one new instance. At the other extreme are methods which involve a single-pass rule formation process, using all available data. There are, of course, many intermediate possibilities. We propose to investigate, within the Meta-DENDRAL framework, whether some of these methods are optimal in the sense of yielding rules of comparatively high quality with the expenditure of comparatively little computing effort. It is hoped that the investigation will lead us to some general insights concerning the optimal utilization of data in automatic rule formation.

Research Plan:

a. Develop and implement one or more procedures for updating an evolving set of rules on the basis of newly examined data. These procedures will make use of existing capabilities of the RULEGEN and RULEMOD programs, and will make possible the implementation of a variety of schemes for data utilization, as described above.

b. Select and implement a representative subset of the class of data utilization schemes indicated above, and test their performance in the application area of mass spectrometry.

c. Describe in a technical report these experiments, their results, and the lessons learned.

Rule Acquisition via Dialogue

Since large stores of knowledge appear to be required for high performance, the process of accumulating that information should be made as easy as possible. The fundamental question here is, how can we make it easy for the expert to tell the system what he knows about the domain. Some initial steps in this direction are described in [Davis76], which reports on the use of what has been labelled "meta-level knowledge" as a basis for establishing communication between the system and an expert. In the simplest terms, meta-level knowledge refers to giving the system the ability to "know what it knows", and can support a wide range of useful abilities.

The basic approach developed there relies on the notion of knowledge acquisition in the context of a shortcoming in the knowledge base. That is, rather than simply asking an expert to "explain all he knows about the field", we allow him to challenge the system with difficult problems and observe its behavior. If he indicates at some point that the system has made a mistake, there is available a large amount of contextual information which can aid in the process of knowledge explication and communication. Thus rather than asking "What is there to know about this domain?", we can say "Here is a problem on which you claim the system made a mistake. Here is the knowledge it used to reach its answer. Now WHAT IS IT THAT YOU KNOW AND THE SYSTEM DOESN'T that allows you to avoid making that mistake?"

This appears to be an effective approach to the problem, since it creates a well defined context, allowing the expert to focus his attempt to describe his knowledge of the domain, and provides the system with a set of expectations about the content of the new knowledge it is going to receive. Both of these offer significant advantages in helping to build up the knowledge base.

Working from this foundation, we plan to extend these ideas to provide a powerful system for knowledge acquisition. Currently, for example, the scope of the context is limited to a particular error in the knowledge base during a particular session with the expert. It ought to be extended to provide a wider perspective, so that the system could form more sophisticated expectations about a particular tutor, thereby making communication between them more effective. Thus rather than forming expectations concerning only the shortcoming presently under examination, for example, the system might be able to consider also the past several shortcomings, in an attempt to detect a broader "theme" in the knowledge it was acquiring.

There ought also to be more effective control over its use of context. The system is currently too "single-minded", in that it holds tenaciously to any expectations it may have formed. There should be a way of indicating to the system that it has formed incorrect assumptions, and that it should "sit back and observe" for a while until it can get "reoriented".

Dealing with large knowledge bases also requires a range of auxiliary capabilities that assist the expert in keeping track of and organizing his work.

Together these constitute a "scratch pad" of sorts that allows him to annotate his new additions, mark existing rules that may need further work, or perhaps examine selected parts of the knowledge base to find areas that may presently be weak. All of these should be aimed at making it possible for the expert to extend his work over several sessions without loss of continuity, and to keep track of both changes that are required and work that has been done, no matter how large the knowledge base may eventually grow to be.

3.3.1.5   GENERAL EXPLANATION SYSTEM

The function of an explanation capability is to permit the user or builder of a knowledge based system to determine:

1. in general, how the system solves problems or uses information;

2. retrospectively, how the system solved a particular problem;

3. interactively, how and why the system came up with its current answers.

The success of the explanation capability for the MYCIN rule based system indicates the usefulness of this capability in debugging the system and in making it easier for a user to learn and believe the system's operations. To make it easier to build explanation capabilities for future knowledge based systems, including systems whose knowledge is embedded in procedures, we intend to construct a system which will provide explanations for a wide class of problem solvers.

Given the appropriate trace of a program's decisions and states, and a model of its problem solving process, it should be possible to answer a variety of well constrained but informative questions about program operation, in general or in a specific run. The aim of this research is to determine what sorts of traces and process models are needed to support selected types of explanations in several classes of knowledge based problem solvers. When the requirements for a class are determined, we intend to implement a general explanation facility to provide the selected explanations for programs in that class. Such a facility should be made useful for several classes of problem solver.

The steps of the research will include:

1. Choose the types of problem solvers to which the explanation system will be applied;

2. Select example knowledge based systems of each class (e.g. protein structure modelling as an example of event/model driven hypothesis formation systems);

3. For each system selected, determine questions to be asked, and what information, such as traces and process descriptions, are needed to answer them;