

request to use MLAB in a small way for modeling. Subsequently the AI potentialities of this domain were recognized by Feigenbaum, Nii, and Osborn and a joint proposal was submitted to and funded by NIH. This past summer John Kunz from Dr. Osborn's laboratory spent approximately half time at Stanford to learn more about AI research and to participate more closely in the development of the PUFF/VM program.

Similarly, Prof. Feigenbaum and Ms. Nii recently spent two days with Profs. Kintsch and Polson at the University of Colorado, introducing them to the newly developed AGE package for use in formulating their program on modeling aspects of human cognition.

A list of the fully authorized projects currently comprising the SUMEX-AIM community can be found with brief abstracts in Appendix A on page 278. More detailed descriptions of collaborative project activities can be found in Section II.

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 providing support for 6 terminals and 4 modems for users as well as a leased line between Stanford and the University of California at Santa Cruz for the Chemical Synthesis project.

### I.D.3 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 project 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 (see Section II.A.1.2 on page 99 for more details).

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. In addition, presentations from a substantial number of outside speakers are invited.

### I.D.4 Existing Project Reviews

We have conducted a continuing careful review of on-going SUMEX-AIM projects to maintain a high scientific quality and relevance to our medical AI goals and to maximize the resources available for newly developing applications projects. At meetings of the AIM Advisory Group and Executive Committee this past year, all the national AIM projects were reviewed.

These groups recommended continued access for most formal projects on the system. However, they recommended that the Higher Mental Functions project could better meet its current goals through computer support at UCLA and we have therefore reduced this project to "associate" status.

#### I.D.5 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 implemented effective system scheduling controls to attempt to maintain the 40:40:20 balance in terms of CPU utilization and to avoid system and user inefficiencies during overload conditions. The initial complement of user projects justifying the SUMEX resource was centered to a large extent at Stanford. Over the past five years 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 although the demand from the national community has increased substantially (see Figure 9 on page 33 and the tables of recent project usage on page 36).

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.) During peak times under the overload control system reported previously, the Stanford community still experiences mutual contentions and delays while the AIM group has relatively open access to the system. We did enable overload controls for the national community this past year, however, because of their substantial increase in demand. For the present, we propose to continue our policy of "soft" allocation enforcement for the fair split of resource capacity.

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.

We have also encouraged mature projects to apply for their own machine resources in order to preserve the SUMEX-AIM resource for research and development efforts and to support projects unable to justify their own machines. The INTERNIST project has received approval for a VAX machine to support their planned development and program testing work. Also Profs. Lesgold and Greeno's "Simulation of Cognitive Processes" project has moved the bulk of their work to their own local VAX.

## I.E Dissemination Efforts

Throughout its existence, SUMEX-AIM has devoted substantial efforts toward disseminating information about its activities as a resource and about the work of individual collaborative projects. We continue to make many presentations at professional meetings, to provide services to demonstrate developed AI programs for interested groups and individuals, and to work in organizing workshops within the SUMEX-AIM community to introduce our work to collaborating professional communities. We have also spent substantial efforts in the past working with the Research Resources Information Center to produce the "Seeds of Artificial Intelligence" monograph to address a broader community of technical and lay people.

The following sections summarize some of the activities undertaken this past year:

### I.E.1 Sixth AIM Workshop

The Sixth Annual AIM (Artificial Intelligence in Medicine) Workshop was held at Stanford University on 13-16 August 1980. The program chairman was Dr. E. Shortliffe, the chairman for demo-based sessions was Dr. L. Fagan, and the short report chairman was Dr. R. Blum. This was the first Workshop to be held in California, and was held in conjunction with the first annual meeting of the AAAI Society (American Association for Artificial Intelligence).

Among the goals of this year's conference was the development of a format for scientific exchange that would help clarify the technical details of the programs that are under development throughout the AIM community. Many individuals have observed that it can be difficult at meetings such as this to obtain detailed understanding of one another's work. Formal presentations with slides and a description of data structures typically are divorced from a sense of the program's operation as seen to the user. As a result, many of us have had to complement our annual Workshop participation with visits to other sites so that we can learn about others' work in depth. In 1980 we experimented with a format that tried to simulate the kind of detailed interactions that have previously occurred only in individual sessions after hours or at times other than the Workshop.

#### Demo-Based Sessions

This year the major portion of the conference was devoted to detailed discussions of AIM systems through the vehicle of specially prepared demonstrations. Each of the established AIM systems was represented with a two hour presentation. Each speaker had a display terminal, special projection system and high-speed connections to the SUMEX 2020 computer. Rather than rely on an impromptu live demonstration, each project was asked to prepare a typescript of an interactive session, subject to the following guidelines:

- (1) The typescript was to represent the interaction exactly as it occurred on the screen to the user (i.e., the presenters were asked not to delete mistakes, problems, garbage collect messages, etc.), and was to be augmented only with the following:
  - (a) annotations to clarify specific points.
  - (b) "break" interruptions as described below.
- (2) The typescript was to be presented in short segments with enough discussion to identify the current point in the program's reasoning process.
- (3) At pertinent points the researchers were asked to break into the program's operation during typescript preparation and display pertinent data structures to illustrate the system's internal representation and organization.

A computer program was written by the SUMEX staff to facilitate the display of annotated and formatted typescripts. The input to the program is a typescript file that has special control characters inserted into text to mark off pages of information. Other control characters are used to highlight (brighten) important points in the typescript, to turn pages or move to a specific page, and to provide for different levels of detail. The provision for different levels of detail was designed to show selectively information in response to questions, or to adjust presentations for different audiences (e.g., physicians vs. computer scientists). Because the program's output is treated as a text file by the system, slide-line material or diagrams can be inserted into the running transcript. A more complete description of the program is available on-line on the SUMEX computer.

No detailed evaluation of the demonstration techniques was undertaken, but our general impression was that the extended speaking time and concentration on program typescripts did orient the talks towards the details of how the programs operate. The major limitations were adequate but less-than-optimal imaging quality from the projection system (particularly in the largest auditorium), and the limited experience of AIM users with the equipment and software used. The SUMEX 2020 with the KI-10's as backup provided excellent computer support for the display technology. One group, the BELIEVER project from Rutgers, augmented their typescript with a "live" demonstration running on the Rutgers AIM resource. The SUMEX staff provided excellent support in the development of programs, equipment setup, and computer support.

A series of 20-minute parallel sessions was also provided for newer AIM systems under development. These talks used standard visual aids. If AIM conferences use demonstration sessions in the future, the featured programs should probably be chosen from among these developing systems.

Since the Workshop, several projects (including MOLGEN, GUIDON, and VM) have used the stored typescript for demonstrations. They have been useful when visitors wish to see a particular program but resources are not

available to run the program in a real-time setting. Each of the demonstration files is available on the SUMEX system, and is available for access by all SUMEX users.

### I.E.2 Tutorial on AI in Medicine

In conjunction with the AIM Workshop, a continuing education tutorial designed for physicians was held at Stanford on August 17-18, 1980. The tutorial was entitled "Computers in Medicine -- Applications of Artificial Intelligence Techniques" and was organized by Drs. W. Clancey and E. Shortliffe. The tutorial had a remarkably good attendance by physicians as well as several other individuals with an interest in the field. The course included an optional introduction to computers for those who had no prior experience with the technology, an overview of SUMEX-AIM research, and an introduction to background materials regarding decision theory and data base applications in medicine. Speakers also provided detailed presentations on MYCIN, CASNET/EXPERT, INTERNIST and GUIDON. The course closed with a panel discussion on the problems and promise of AI in Medicine. It was accredited for postgraduate medical education through Stanford University School of Medicine; the 135 physicians in attendance earned 11.5 continuing education credits. In addition, 18 students, several non-physician researchers, and 10 members of the press attended. Enrollees came from as far away as Mexico and the East Coast. For the reasonable fee of \$40 covering the two days of lectures, the attendees also received a syllabus of readings and two lunches.

The syllabus is a comprehensive survey of medical AI research and is comprised of recent articles written by the tutorial faculty, mostly for a clinical audience. The faculty consisted of 15 distinguished researchers from the AIM community, including 7 physicians and 9 speakers from centers other than Stanford. By holding the tutorial immediately after the AIM Workshop and before the first Annual Meeting of the American Association for Artificial Intelligence (AAAI), we were able to secure the participation of expert physicians in the field who were already at Stanford (Drs. Greenes, Lindberg, Myers, and Pauker), as well as computer scientists from the East Coast (Drs. Davis, Kulikowski, Pople, Szolovits, and Swartout). Stanford speakers included Drs. Blum, Buchanan, Clancey, Feigenbaum, Fries, and Shortliffe. Coordination and planning for the tutorial was facilitated by sending electronic messages; almost all speakers regularly use SUMEX or another ARPANET machine.

To evaluate the impact of the tutorial on the participants, and to assess baseline opinions regarding the field, we undertook a survey of the physicians' knowledge about computers as well as their attitudes towards medical consultation systems. The statistical analysis of these questionnaires has now been completed, and a paper summarizing the results submitted for publication (Teach, R.L. and Shortliffe, E.H. "An analysis of physician attitudes regarding computer-based clinical consultation systems." Submitted for publication, March 1981). In brief, the survey showed that physicians were willing to accept the possibility of computer-based clinical decision aids but placed severe demands on the capabilities of such systems if they were to be acceptable for routine use.

In addition, attendees were asked to evaluate the course itself, as well as the talks by individual speakers. These forms showed that the course was exceedingly well received. Attendees were fascinated by the content, generally felt it was well presented, and indicated they would

recommend the course to others if it were made available again. Many physicians requested a follow-up course that would introduce them to more technical detail than had been possible in the introductory tutorial.

In conclusion, we believe that the tutorial was an encouraging success, and demonstrated the effectiveness of this kind of forum for introducing physicians to the research efforts within the AIM community. The faculty is enthusiastic about repeating the course, possibly on the East Coast in conjunction with a future AIM Workshop. Several members of the audience expressed interest in detailed, small group discussions of particular AIM programs. We believe these discussions could be a valuable way of exporting our methods and approach beyond the immediate AIM community.

### I.E.3 GENET - An Experiment in AI System Dissemination

#### Background

The MOLGEN project at Stanford (see Section II.A.1.5 on page 136) has focused on applications of artificial intelligence and symbolic computation to the field of molecular biology. The research began in 1975 and is currently in the first year of a three year grant renewal. In early 1980 it was realized that some of the systems developed by MOLGEN were of direct utility to many scientists in the domain. Accordingly, with the cooperation of the SUMEX-AIM staff and close coordination with the AIM Executive Committee, it was decided in February 1980 to provide a carefully limited guest service for the community use of such systems.

There were two major reasons for the establishment of this guest service, which took the form of the GENET account on SUMEX. The first was to broaden MOLGEN's base of scientist collaborators, to find molecular biologists at institutions other than Stanford who could contribute actively to our knowledge-based approach to problem solving. The second was to introduce a generally computer-naive community to the benefits of resource sharing provided by a system like SUMEX, with the hope of serving as a model for the dissemination of other AI software and possibly for an eventual resource for molecular biology.

We believe that we have succeeded in these two goals. Many of our GENET guests have become active collaborators in core MOLGEN research. These collaborators include Professor Allan Maxam at Harvard Medical School, Dr. Walter Goad at Los Alamos, Dr. Richard Roberts at Cold Spring Harbor, Dr. William Pearson at Johns Hopkins, Drs. Walter Bodmer, Julia Bodmer, and Robert Kamen at the Imperial Cancer Research Fund, Professor Fred Blattner at Wisconsin, Dr. Andrew Taylor at University of Oregon, and Dr. Dan Davison of SUNY-Stonybrook. We are also pleased by the numerous comments SUMEX has received from GENET users praising the user-sensitive nature of the resource, especially in comparison to typical university computer centers.

GENET has been important both for MOLGEN and for the national community of molecular biology. It has ensured a steady flow of ideas for the artificial intelligence research that is core to both the MOLGEN grant and the SUMEX-AIM mission. It has also provided a useful service to an international community that is not readily available elsewhere.

#### GENET Community Management

Our decision to support the GENET guest experiment and our approach to doing so within the SUMEX-AIM resource has been reviewed and approved both by the AIM Executive Committee and by the Initial Review Group/National Advisory Research Resources Council in the course of the peer review of our pending SUMEX renewal application. We have tried to manage the GENET guest experiment in such a way that we maintain the "friendly" interface of the SUMEX-AIM resource for molecular biologists unfamiliar with computers while taking appropriate steps so that GENET

usage does not detract from on-going AI research and so that we assure prudent administration SUMEX as an NIH-BRP resource. The key elements in our management approach include:

- 1) Controlled announcement of the GENET opportunity -- Beginning in February 1980, the availability of GENET services was announced, primarily by talks at professional conferences with accompanying program demonstrations. We decided against publishing "blanket" announcements in professional journals in order to maintain a very high standard of collaborator interest and scientific expertise within the limited group we could serve with available SUMEX resources.
- 2) Close coordination with the AIM Executive Committee -- We kept the AIM Executive Committee apprised of plans for the GENET experiment and of progress and growth of the community. At the August 1980 AIM Workshop meeting of the Executive Committee, Professor L. Kedes of the MOLGEN project made a presentation on the status of GENET. The Executive Committee approved continuation of the GENET service but because of the significant growth in the number of GENET users and their consumption of CPU resources, a limit of two simultaneous GENET jobs was placed on the community. The Executive Committee also approved the concept of a proposed Molecular Biology Computing Resource related to but separate from the existing SUMEX resource.
- 3) Careful control of GENET usage -- We have closely monitored the very rapid growth in GENET usage of SUMEX (see data below). With Executive Committee advice and in cooperation with the MOLGEN project personnel managing the GENET community, we have instituted several successively stringent controls on GENET users:
  - a) All GENET users run out of the same directory so scheduler control limits are enforced to hold GENET usage as a whole down relative to that of AI research projects during heavy loads.
  - b) The GENET directory has been intentionally limited in disk space allocation so that large numbers of files cannot be retained.
  - c) Starting in October 1980, a limit of two simultaneous logged-in GENET jobs was placed on the community.
  - d) Starting in December 1980, a policy statement was issued restricting GENET use to academic collaborators. MOLGEN project management informed industrial collaborators that they could no longer use the GENET facility and actively monitored adherence to this policy. Previously, valuable feedback had been obtained from a small group of industrial collaborators for MOLGEN AI program development. However, with the rapid growth of the highly competitive molecular genetics industry, there was no way we could adequately control industrial users consistent with SUMEX's status as a federally funded national resource. Thus, we decided to exclude them. In April 1981, we instituted a GENET user password checking system to further control community access, particularly in regard to industrial users.

- 4) Limited commitment of SUMEX staff resources -- The day to day management of the GENET community has been the responsibility of MOLGEN project personnel. SUMEX personnel have only contributed to developing system facilities to help manage GENET (guest and GENET password capabilities), assisted with technical communications problems, and advised in establishing GENET management policies consistent with AIM Executive Committee and SUMEX Principal Investigator resource policies. The total commitment of staff time has been on the order of 1-2 man-months.

#### Scope of the GENET User Community

The GENET community consists of approximately 200 users from 63 research institutions. Of these 200 users, approximately 35 are consistently active users. That is, they log in, run programs, and interact with the MOLGEN members on an almost daily basis. Many of these users have made valuable contributions to our work. About 100 others are frequent, but not regular users. They log in only when they have a major analysis task to perform, which seems to be on the order of once a month.

The remaining users rarely use the system. They have logged in a few times, but for one reason or another they never become regular users of the system. Quite often this is because a lab group will settle on having one or two graduate students or post-doctoral associates become the "computer experts" of the group, and as a result, the computer use by the other people in the lab drops to an almost non-existent level. Unfortunately, an equally prevalent reason for users to stop using the GENET account is a lack of resource time. Probably the major complaint that we get from GENET users is concerning the lack of compute time and availability of the system. One account just is not enough for 200 people to share, especially when it is restricted to 2 jobs at one time. We constantly remind the GENET users to use there resources wisely. We encourage them to use the BATCH system to run job in the wee hours of the morning, and we remind them to be prepared to do their work quickly when they log in to the system, but their efforts do not seem to help the problem very much.

Most GENET users use only a small set of programs. These consists of text editors, which are used to set up the data files that for the MOLGEN analysis programs; XSEARCH, which GENET users use to effectively search through our database for sequences that can assist them in their research; and the electronic mail facilities. Very few of our GENET users actually feel comfortable using programs other than the ones that we maintain, not because the other programs would not be useful, but instead because the users do not have the computer time to experiment with what is available.

There are three note-worthy programs that we provide for GENET users that are used extensively. SEQ, a DNA-RNA sequence analysis program, which is continually being improved, is the most widely used. MAP, a program that assists in the construction of restriction maps from restriction enzyme digest data, is also used a great deal. Finally, a new program, MAPPER (written and maintained by William Pearson from Johns Hopkins University), is a simplified version of the MOLGEN MAP program that is somewhat more

efficient than the MOLGEN version. The MOLGEN UE program and special molecular genetics knowledge bases are not available to the general GENET user at this time for two reasons. First of all, the UE program is quite costly to use (in terms of computer cycles), and secondly, we feel that the knowledge base is not quite ready for the computer novice to learn and use without a significant amount of initial assistance. A few GENET users (mostly Stanford associates) that have had a significant interest in the knowledge base have become EXO-MOLGEN users and are developing knowledge bases on their own which we hope will eventually be added to the ones that MOLGEN is developing and maintaining.

#### GENET Usage Statistics

Following is a table of monthly statistics for GENET usage of SUMEX. Note "TOTAL CONNECT HOURS" includes connect time for local dialups, hardlines, ARPANET, and TYMNET. "TYMNET CONNECT HOURS" includes that part of the total connect time which is via TYMNET and for which SUMEX pays a separate usage charge. Recent GENET TYMNET usage has been about 20-25% of the total SUMEX TYMNET connect time. Our monthly TYMNET bills are about \$5,000, so monthly GENET TYMNET usage is about \$1,125. Most GENET users come from other parts of the country and no additional local dial-up lines have been installed to support GENET usage.

Month/ Year	CPU Hours	Total Connect Hours	TYMNET Connect Hours	GENET % of Sumex TYMNET Use	File Pages
Feb/80	3.23	32.72	18.88	2.0%	57
Mar/80	1.28	51.57	12.80	1.4	95
Apr/80	8.37	117.87	51.73	5.4	209
May/80	9.20	104.46	66.65	8.0	166
Jun/80	11.08	188.35	118.03	11.7	253
Jul/80	19.21	342.87	189.00	18.2	231
Aug/80	18.71	257.23	188.53	18.2	367
Sep/80	57.32	409.83	254.53	28.5	626
Oct/80	36.47	348.66	211.95	23.3	920
Nov/80	82.90	648.56	308.40	31.1	1133
Dec/80	19.86	295.85	188.67	22.8	1110
Jan/81	48.00	747.91	277.30	27.2	996
Feb/81	22.58	265.39	163.55	16.1	962
Mar/81	29.73	613.74	313.57	25.0	982
Apr/81	43.04	662.57	unavail	unavail	1633

Plots of the CPU usage, connect time, and file usage data can be found in Figures 16-18.

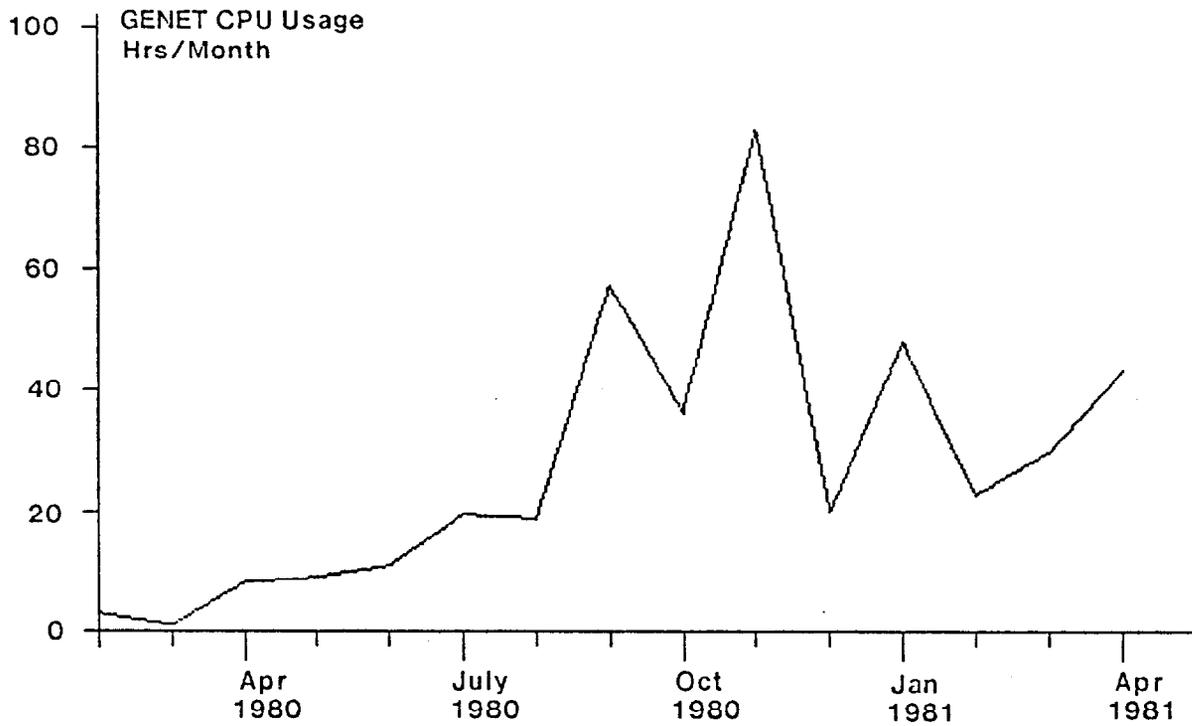


Figure 16. GENET CPU Usage by Month

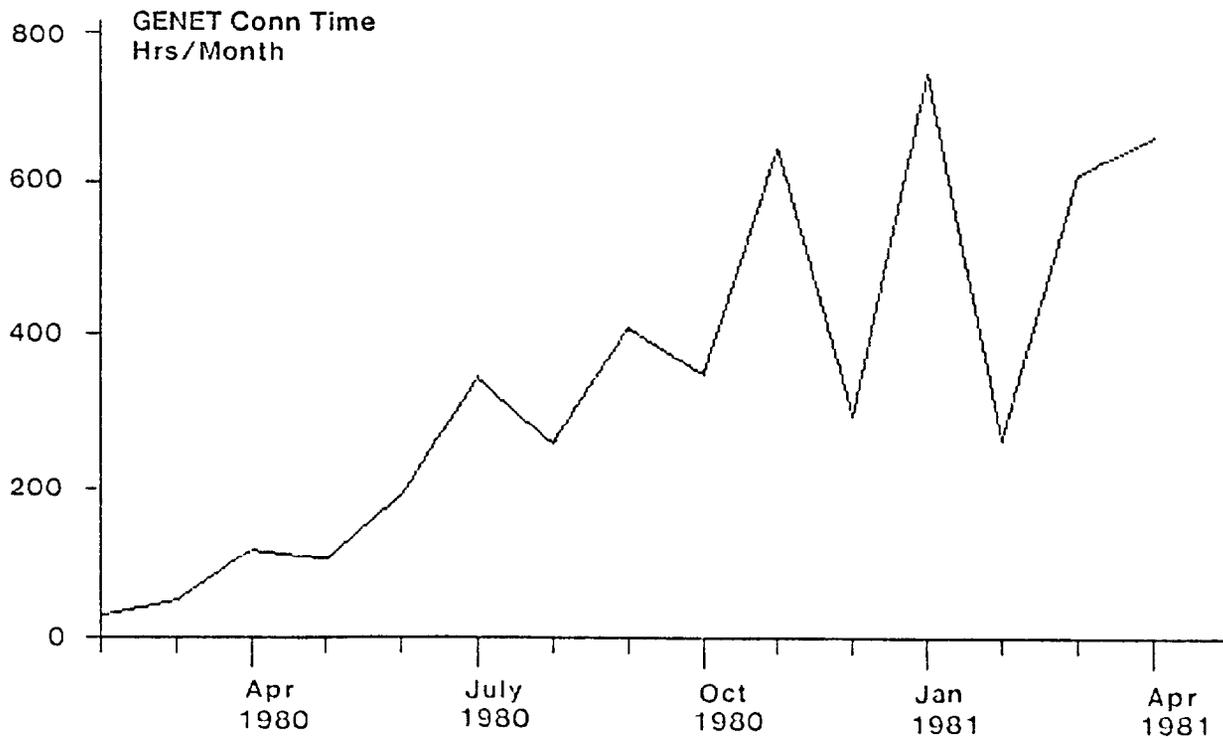


Figure 17. GENET Connect Time by Month

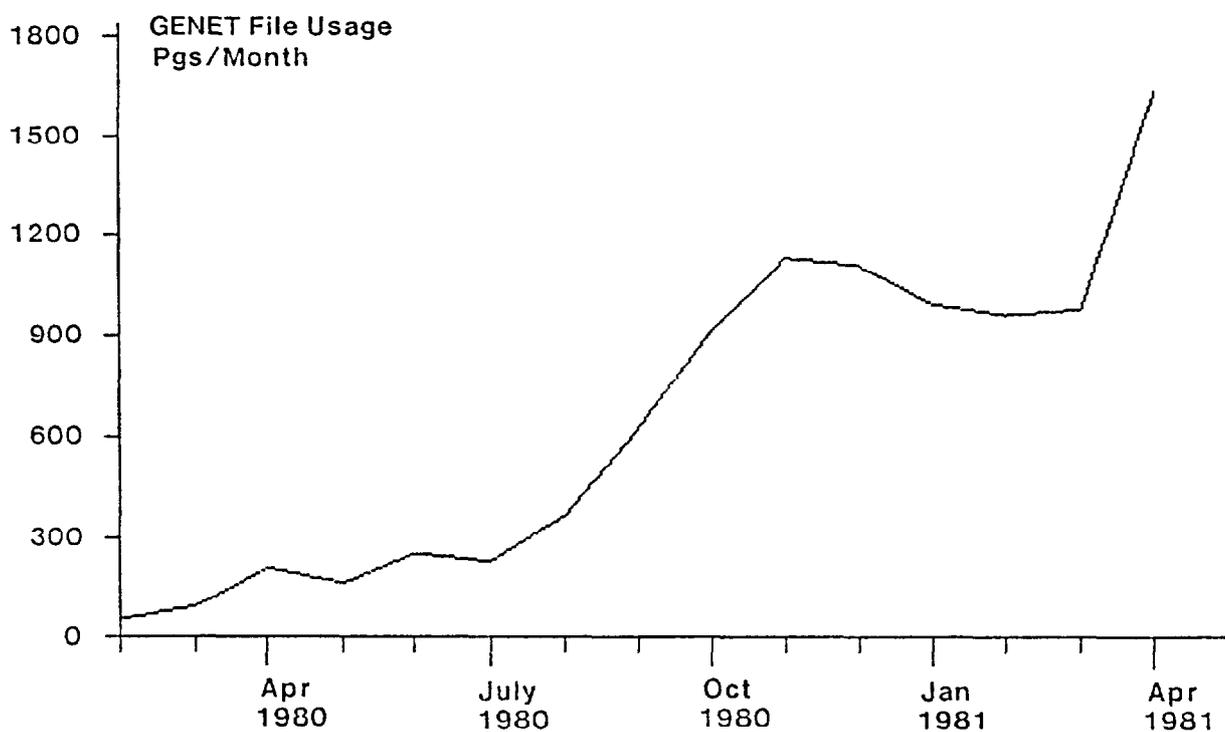


Figure 18. GENET File Space by Month

I.F Comments on the Biotechnology Resources ProgramResource Organization

We firmly believe that the Biotechnology Resources Program is one of the most effective vehicles for developing and disseminating technological tools for biomedical research. The goals and methods of the program are well-designed to encourage building of the necessary multi-disciplinary groups, merging appropriate technological and medical disciplines. In our experience with the SUMEX-AIM resource, several elements of this approach seem to emerge as key to the development and management of an effective resource:

- 1) Effective Management Framework - there needs to be an explicit agreement between the BRP and the resource principal investigator that sets out a clear mandate for the resource and its allocation, provides worthwhile incentives for the host institution and investigator to invest the necessary substantial professional career time to develop and manage the resource, and ensures equitable distribution of resource services to its target community.
- 2) Close Working Relationship with NIH - a resource is a major and often long-term investment of money and human energy. A close and mutually supportive working relationship between resource management, its advisory committees, and the NIH administration is essential to assure healthy development of the resource and its relationship to its user community. We at SUMEX-AIM have benefited immensely from such a relationship with Dr. William R. Baker, Jr. in the evolution of the SUMEX-AIM community.
- 3) Freedom to Explore Resource Potential - a resource, by its nature, operates at the "cutting edge" in developing its characteristic technology and learning how to effectively disseminate it to the biomedical community at large. BRP should not impose artificial constraints on the resource for commercializing its efforts (fees for service) or developing its potential (budget ceilings). Such artificial policy impositions can serve to undermine the very goals central to BRP's reason for existence. Satisfactory policies in this regard have been worked out recently and should be retained.

Electronic Communications

SUMEX-AIM has pioneered in developing more effective methods for facilitating scientific communication. Whereas face to face contacts continue to have their place, in the longer term we feel that computer-based communications will become increasingly important to NIH and the biomedical community. We would like to see BRP take a more active role in promoting these tools within NIH and its grantee community. A concrete step would be to become a sponsoring agency for the ARPANET which remains the most effective means for a very broad spectrum of services to promote good communications. This could serve as a base for interconnecting sponsored machines and offering a broader range of services and promoting broader collaboration among the biomedical community at large.

## II Description of Scientific Subprojects

### II.A Scientific Subprojects

The following subsections report on the AIM community of projects and "pilot" efforts including local and national users of the SUMEX-AIM facility at Stanford. Those using the Rutgers-AIM facility are annotated with "[Rutgers-AIM]". In addition to these detailed progress reports, we have included briefer summary abstracts of the fully authorized projects in Appendix A on page 278.

The collaborative project reports and comments are the result of a solicitation for contributions sent to each of the project Principal Investigators requesting the following information:

#### I. SUMMARY OF RESEARCH PROGRAM

- A. Project rationale
- B. Medical relevance and collaboration
- C. Highlights of research progress
  - Accomplishments this past year
  - Research in progress
- D. List of relevant publications
- E. Funding support (see details below)

#### II. INTERACTIONS WITH THE SUMEX-AIM RESOURCE

- A. Medical collaborations and program dissemination via SUMEX
- B. Sharing and interactions with other SUMEX-AIM projects  
(via computing facilities, workshops, personal contacts, etc.)
- C. Critique of resource management  
(community facilitation, computer services, communications services, capacity, etc.)

#### III. RESEARCH PLANS (8/80-7/86)

- A. Project goals and plans
  - Near-term
  - Long-range
- B. Justification and requirements for continued SUMEX use
- C. Needs and plans for other computing resources beyond SUMEX-AIM
- D. Recommendations for future community and resource development

We believe that the reports of the individual projects speak for themselves as rationales for participation; in any case the reports are recorded as submitted and are the responsibility of the indicated project leaders.

II.A.1 Stanford Projects

The following group of projects is formally approved for access to the Stanford aliquot of the SUMEX-AIM resource. Their access is based on review by the Stanford Advisory Group and approval by Professor Feigenbaum as Principal Investigator.

II.A.1.1 AGE - Attempt to Generalize

## AGE - Attempt to Generalize

H. Penny Nii and Edward A. Feigenbaum  
Computer Science Department  
Stanford University

ABSTRACT: Isolate inference, control, and representation techniques from previous knowledge-based programs; reprogram them for domain independence; write an interface that will help a user understand what the package offers and how to use the modules; and make the package available to other members of the AIM community and labs doing knowledge-based programs development, and the general scientific community.

I. SUMMARY OF RESEARCH PROGRAM

## A. Project Rationale

The general goal of the AGE project is to demystify and make explicit the art of knowledge engineering. It is an attempt to formulate the knowledge that knowledge engineers use in constructing knowledge-based programs and put it at the disposal of others in the form of a software laboratory.

The design and implementation of the AGE program is based primarily on the experience gained in building knowledge-based programs at the Stanford Heuristic Programming Project in the last decade. The programs that have been, or are being, built are: DENDRAL, meta-DENDRAL, MYCIN, HASP, AM, MOLGEN, CRYSLIS [Feigenbaum 1977], and SACON [Bennett 1978]. Initially, the AGE program will embody artificial intelligence methods and techniques used in these programs. However, the long-range aspiration is to integrate those developed at other AI laboratories. The final product is to be a collection of building-block programs combined with an "intelligent front-end" that will assist the user in constructing knowledge-based programs. It is hoped that AGE will speed up the process of building knowledge-based programs and facilitate the dissemination of AI techniques by: (1) packaging common AI software tools so that they need not be reprogrammed for every problem; and (2) helping people who are not knowledge engineering specialists write knowledge-based programs.

## B. Medical Relevance and Collaboration

AGE is relevant to the SUMEX-AIM Community in two ways: as a vehicle for disseminating cumulated knowledge about the methodologies of knowledge engineering and as a tool for reducing the amount of time needed to develop knowledge-based programs.

(1). Dissemination of Knowledge: The primary strategy for conducting AI research at the Stanford Heuristic Programming Project is to build complex programs to solve carefully chosen problems and to allow the

problems to condition the choice of scientific paths to be explored. The historical context in which this methodology arose and summaries of the programs that have been built over the last decade at HPP are discussed in [Feigenbaum 1977]. While the programs serve as case studies in building a field of "knowledge engineering," they also contribute to a cumulation of theory in representation and control paradigms and of methods in the construction of knowledge-based programs.

The cumulation and concomitant dissemination of theory occur through scientific papers. Over the past decade we have also cumulated and disseminated methodological knowledge. In Computer Science, one effective method of disseminating knowledge is in the form of software packages. Statistical packages, though not related to AI, are one such example of software packages containing cumulated knowledge. AGE is an attempt to make yesterday's "experimental technique" into tomorrow's "tool" in the field of knowledge engineering.

(2). Speeding up the Process of Building Knowledge-based Programs: Many of the programs built at HPP are intelligent agents to assist human problem solving in tasks of significance to medicine and biology (see separate sections for discussions of work and relevance). Without exception the programs were handcrafted. This process often takes many years, both for the AI scientists and for the experts in the field of collaboration.

AGE will reduce this time by providing a set of preprogrammed inference mechanisms and representational forms that can be used for a variety of tasks. Close collaboration is still necessary to provide the knowledge base, but the system design and programming time of the AI scientists can be significantly reduced. Since knowledge engineering is an empirical science, in which many programming experiments are conducted before programs suitable for a task are produced, reducing the programming and experimenting time would significantly reduce the time required to build knowledge-based programs.

### C. Highlights of Research Summary

Last year we reported the addition of Backchaining framework (the chaining of production rules in the manner similar to that used in MYCIN) and an interface to the Units package (for additional representational form and its use from AGE rules). In the past year we placed our research emphasis on (1) improving the existing component parts and the user interface, (2) developing debugging facilities, and (3) producing additional documents.

We completed the implementation of Trace and Break packages, as well as a facility for trace-back explanation. Using the trace-back facility users can inquire about the program's actions; AGE answers the questions by using the execution history list. Some example questions are: "What was the hypothesis before the execution of rule 2 in KS X?", "What Event led to the activation of KS X?". Since AGE has no knowledge of the application domain, it cannot "explain" the program actions in the language of the domain, but it produces "explanations" that are useful to the implementers.

We found that the specification and editing protocols for the various components were awkward and difficult for the users to learn. We redesigned this particular portion of the interface and have completed about 75% of the re-implementation.

In addition to the standard documents (a user's guide and a reference manual), we began a documented series of examples. These examples are actually implemented and running programs; each document consists of a description of the example problem, its formulation in terms of AGE, reasons for the particular formulation, and a complete program listing. In addition, the programs are available for the users to run. We observed that our documents, like most other program documentations, are useful only to those people who are already familiar with AGE. The Example Series is an experiment to see if a combination of standard documents and examples would be of any significant help to new users.

#### D. Publications

Nii, H. Penny and Aiello, Nelleke, "AGE: a knowledge-based program for building knowledge-based programs," Proc. of IJCAI-6, pp. 645-655, vol. 2, 1979.

Nii, H. Penny, "An Introduction of Knowledge Engineering, Blackboard Model, and AGE," HPP Working Paper, HPP-80-29.

Aiello, N. and Nii, H.P., "The Joy of AGE-ing: A User's Guide to AGE-1."

Aiello, N., Bock, C., Nii, H.P., White, W., "AGE Reference Manual."

AGE Example Series 1: "BOWL: A Beginner's Program."

AGE Example Series 2: "AGEPUFF: A Simple Event-Driven Program."

## II. INTERACTION WITH THE SUMEX-AIM RESOURCES

### AGE Availability:

Currently AGE-1 is available to a limited number of groups on the PDP-10 at the SUMEX-AIM Computing Facility and on the PDP-20/60 at the SCORE Facility of the Computer Science Department. The current implementation is described briefly in a later section.

### Dissemination:

We previously reported a three-day workshop that we conducted in March 1980. The aims of the workshop were to familiarize the attendees with the use of AGE, and for each participant to implement a running program related to his application area. Of the attendees, the group from the Institute of Medical Electronics, University of Tokyo, has continued to use AGE to develop a medical diagnosis program.

In addition, many of the activities of the past year described earlier were direct results of what we learnt at the workshop.

For the 1980 AIM Workshop we reimplemented in AGE a major portion of the VM program (described elsewhere). In addition to demonstrating a variety of features of AGE, we were able to demonstrate the relatively short implementation time required once the goals of the application and the necessary knowledge were delineated -- a first-year graduate student had the program running in three weeks.

#### Profile of the Current AGE System:

To correspond to the two general technical goals described earlier, AGE is being developed along two separate fronts: the development of tools and the development of "intelligent" user interface.

#### Currently Implemented Tools:

The current AGE system provides the user with a set of preprogrammed modules called "components" or "building blocks". Using different combinations of these components, the user can build a variety of programs that display different problem-solving behavior. AGE also provides user interface modules that help the user in constructing and specifying the details of the components. A component is a collection of functions and variables that support conceptual entities in program form. For example, production rule, as a component, consists of: (1) a rule interpreter that support the syntactic and semantic description of production-rule representation as defined in AGE, and (2) various strategies for rule selection and execution.

The components in AGE have been carefully selected and modularly programmed to be useable in combinations. For those users not familiar enough to experiment with combining the components, AGE currently provides the user two predefined configuration of components--each configuration is called a "framework". One framework, called the Blackboard framework, is for building programs that are based on the Blackboard model [Lesser 77]. Blackboard model uses the concepts of a globally accessible data structure called a "blackboard", and independent sources of knowledge which cooperate to form hypotheses. The Blackboard model has been modified to allow flexibility in representation, selection, and utilization of knowledge. The other framework, called the Backchain framework, is for building programs that use backward-chained production rules as its primary mechanism of generating inferences.

#### The Front-End:

To support the user in the selection, specification, and use of the components, AGE is currently organized around four major subsystems that interact in various ways. Around it is a system executive that allows the user access to the subsystems through menu selection. Figure 1. shows the general interrelationship among these subsystems.

The Browse and Design subsystems help to familiarize the user with AGE and to guide the user in the construction of his programs through the use of predefined frameworks. The third subsystem is a collection of interface modules that help the user specify the various components of the framework. The last subsystem is designed for testing and refining the user program. Each of the subsystem is described in more detail below:

**BROWSE:** The function of Browse subsystem is to guide the user in browsing through its textual knowledge base, called the MANUAL. The MANUAL contains (a) a general description of the building-block components on the conceptual level; (b) a description of the implementation of these concepts within AGE; (c) a description of how these components are used within the object program; (d) how they can be constructed by the user; and (e) various examples. The information in the MANUAL is organized to represent the conceptual hierarchy of the components and to represent the functional relationship among them.

**DESIGN:** The function of the DESIGN subsystem is to guide the user in the design and construction of his program through the use of predefined configuration of components, or framework. Each framework is defined in DESIGN-SCHEMA, a data structure in the form of AND/OR tree, that, on one hand, represents all the possible configuration of components within the framework; and, on the other hand, represents the decisions the user must make in order to design the details of the user program. Using this schema, the DESIGN subsystem guides the user from one design decision point to another. At each decision point, the user has access to the MANUAL and also to advice regarding design decisions at that point. An appropriate ACQUISITION module can be invoked from the DESIGN subsystem so that general design and implementation specifications can be accomplished simultaneously.

**ACQUISITION:** For each component that the user must specify, there is a corresponding acquisition/editor module that queries the user for task-specific information. The calling sequence of the acquisition module is guided by DESIGN-SCHEMA when the user is using the DESIGN subsystem. They can also be accessed directly from the system menu or Interlisp.

**INTERPRETER:** This subsystem contains several modules that help the user run and debug his program. The Check module checks for the completeness and correctness of the specification for an entire framework. The Interpreter executes the user program. The Trace and Break modules are run-time debugging aids. The Editor, Check, Trace, Break, and the Explanation (described below) modules are designed to complement each other, and to help the user observe the workings of his program and to make corrections as necessary.

**EXPLANATION:** AGE has enough information to replay its execution steps, and it has reasonable justifications for the actions within the various framework. AGE provides a back-trace explanation facility whereby questions related to the execution history can be answered by the system interactively. However, AGE is totally ignorant of the user's task domain and has no means of conducting a dialogue about the specifics of the domain. A detailed history of the execution steps is available to the user to build his own domain specific explanation, if necessary.

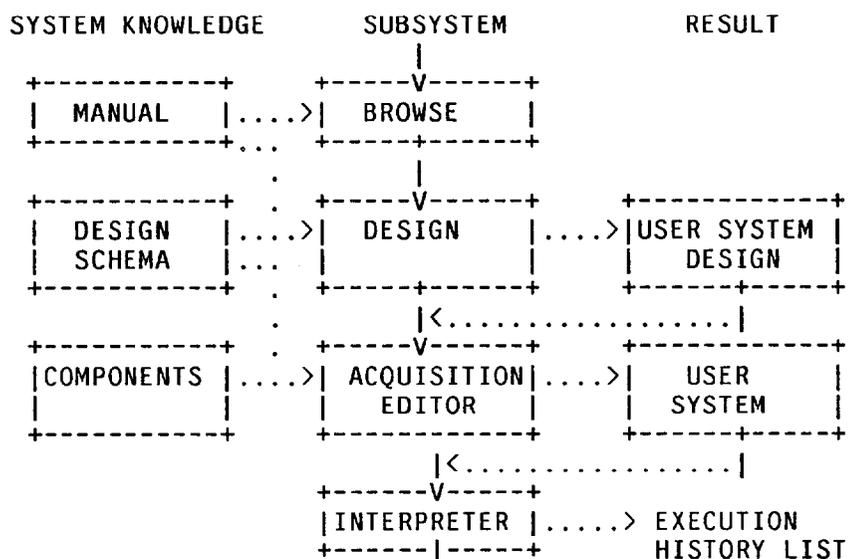


Figure 1. AGE System Organization  
(... = data flow; --- = control flow)

### III. RESEARCH PLAN

#### Research Topics:

The task of building a software laboratory for knowledge engineers is divided into two main sub-tasks:

1. The isolation of techniques used in knowledge-based programs: 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 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. One of our goals is to isolate the AI techniques that are general and determine precisely the conditions for their use.

2. Guiding the user in the initial application of these techniques: Once the various techniques are isolated and programmed for use, an intelligent agent is needed to guide the user in the application of these techniques. In AGE-1, we assume that the user understands AI techniques, knows what she wants to do, but does not understand how to use the AGE system to accomplish his task. A longer range interest involves helping the user determine what techniques are applicable to his task, i.e. it will assume that the user does not understand the necessary techniques of writing knowledge-based programs.

Research Plan:

In our judgement the the first research task has progressed enough to a point where can continue on to the second task. The system that embodies the results to date is called AGE-1. The structure of AGE-1 is now frozen and only minor modifications are being made. We will continue to support it by correcting bugs and adding requested features that are easily implementable.

## AGE-2

AGE-2 will try to address the second of the research tasks described above.

Although the current Design subsystem provides specification functions that allow the user to interactively specify the knowledge of the domain and the control structure, it does not (aside from simple advise) provide the user any help in the actual design process. For example, AGE should be able to provide some aids to the user on what kinds of inference mechanisms and representations are appropriate for his application problem. We have stated this problem in our previous reports without any promising ideas on how we might attack this problem. With the variety of feedbacks we received from our experimental users, we now understand a few of the problems the inexperienced users are faced with. With these in mind, we have begun, and will continue, to explore ways in which we can redesign and add facilities that will help users who are not familiar with knowledge engineering techniques and methodologies.

One of the major obstacles in the way of AGE-2 development is the way in which AGE-1 is implemented. Although the syntax of AGE-1 is clearly defined (see the Reference Manual), the semantics are not well-defined. They are defined in ad hoc fashion in the Editor, the Interpreter, and the Check modules. In order for AGE-2 to be able to conduct a dialogue about itself with the user, its semantics, as well as its syntax, must be uniformly represented. Since very little research results are available in the area of representing the semantics of systems (one exception is in the automatic programming research), we need to experiment with a variety of approaches. We have already begun to look into some alternative representations. In changing the representation of the AGE system, no new components will be added, and minimum amount of changes will be made to the definition of the existing components.

Concurrent with re-representing the AGE system, we will identify a dozen or so framework, in addition of the existing two, that have simpler constructs and are easier for the novice users to understand. The simplicity will be achieved by providing less options for the user -- options which, because of their nature, are confusing to new users. Limiting the degrees of freedom for the user has the side benefit of allowing AGE to provide more specific description and aids. For example, in a very constrained framework we can provide a library of "standard" predicates for the users, which can have associated with them English translations; with such texts available the rules and the back-trace explanation can be printed in English-like form. Once the user is

comfortable with the more simple frameworks, he can add complexity simply by replacing the predefined options selected for the frameworks.

Computing Resources and Management:

We believe the computing and communication resources provided by the SUMEX Facility make it one of the best in the country. The management is responsive to the needs of the research community and provides superb services. However, the system is getting to a point where no serious research and development is possible, because of the lack of computing cycles due to overcrowding. It is a compliment to the facility that there are so many users. On the other hand, our productivity has gone down in recent months, because of the heavy load on the system. It would appear that the situation will not improve on its own, since many of the projects that were small a few years ago are maturing into larger, more complex systems. Which is the way it should be. The environment in which the work is done also needs to grow. In short, without augmentation to the current computing power and storage space (which had never been generous), our ability to make research progress at SUMEX will be drastically curtailed.

II.A.1.2 AI Handbook Project

## Handbook of Artificial Intelligence

E.A. Feigenbaum, A. Barr, and P. Cohen  
Stanford Computer Science Department

I. SUMMARY OF RESEARCH PROGRAM

## A. Technical Goals

The AI Handbook is a compendium of knowledge about the field of Artificial Intelligence. It is being compiled by students and investigators at several research facilities across the nation. The scope of the work is broad: Two hundred articles cover all of the important ideas, techniques, and systems developed during 20 years of research in AI. Each article, roughly four pages long, is a description written for non-AI specialists and students of AI. Additional articles serve as Overviews, which discuss the various approaches within a subfield, the issues, and the problems.

There is no comparable resource for AI researchers and other scientists who need access to descriptions of AI techniques like problem solving or parsing. The research literature in AI is not generally accessible to outsiders. And the elementary textbooks are not nearly broad enough in scope to be useful to a scientist working primarily in another discipline who wants to do something requiring knowledge of AI. Furthermore, we feel that some of the Overview articles are the best critical discussions available anywhere of activity in the field.

To indicate the scope of the Handbook, we have included an outline of the articles as an appendix to this report (see page 303).

## B. Medical Relevance and Collaboration

The AI Handbook Project was undertaken as a core activity by SUMEX in the spirit of community building that is the fundamental concern of the facility. We feel that the organization and propagation of this kind of information to the AIM community, as well as to other fields where AI is being applied, is a valuable service that we are uniquely qualified to support.

## C. Progress Summary

Because our objective is to develop a comprehensive and up-to-date survey of the field, our article-writing procedure is suitably involved. First drafts of Articles are reviewed by the staff and returned to the author (either an AI scientist or a student in the area). His final draft is then incorporated into a Chapter, which when completed is sent out for review to one or two experts in that particular area, to check for mistakes and omissions. After corrections and comments from our reviewers are

incorporated by the staff, the manuscript is edited, and a final computer-prepared, photo-ready copy of the Chapter is generated.

We expect the Handbook to reach a size of approximately 1000 pages. Roughly two-thirds of this material will constitute Volumes I and II of the Handbook. The material in Volumes I and II will cover AI research in Heuristic Search, Representation of Knowledge, AI Programming Languages, Natural Language Understanding, Speech Understanding, Automatic Programming, and Applications-oriented AI Research in Science, Mathematics, Medicine, and Education. Researchers at Stanford University, Rutgers University, SRI International, Xerox PARC, RAND Corporation, MIT, USC-ISI, Yale, and Carnegie-Mellon University have contributed material to the project.

#### D. List of Relevant Publications

Many of the chapters of Volumes I and II of the AI Handbook have already appeared in preliminary form as Stanford Computer Science Technical Reports, authored by the respective chapter-editors. References follow. Other chapters of Volumes II and III will appear as Technical Reports in the summer and fall of 1981.

HPP-79-12 (STAN-CS-79-726)  
Ann Gardner. Search.

HPP-79-17 (STAN-CS-79-749)  
William Clancey, James Bennett, and Paul Cohen.  
Applications-oriented AI Research: Education.

HPP-79-21 (STAN-CS-79-754)  
Anne Gardner, James Davidson, and Terry Winograd.  
Natural Language Understanding.

HPP-79-22 (STAN-CS-79-756)  
James S. Bennett, Bruce G. Buchanan, and Paul R. Cohen.  
Applications-oriented AI Research: Science and Mathematics.

HPP-79-23 (STAN-CS-79-757)  
Victor Ciesielski, James S. Bennett, and Paul R. Cohen.  
Applications-oriented AI Research: Medicine.

HPP-79-24 (STAN-CS-79-758)  
Robert Elschlager and Jorge Phillips. Automatic Programming.

HPP-80-3 (STAN-CS-80-793)  
Avron Barr and James Davidson. Representation of Knowledge.

#### E. Funding Support Status

The Handbook Project is partially supported under the Heuristic Programming Project contract with the Advance Research Projects Agency of the DOD, contract number MDA 903-77-C-0322, E. A. Feigenbaum, Principal Investigator and under the core research activities of the SUMEX-AIM resource.