# SUMEX

STANFORD UNIVERSITY
MEDICAL EXPERIMENTAL COMPUTER RESOURCE
RR-00785

ANNUAL REPORT—YEAR 15

Submitted to

OMEDICAL RESEARCH TECHNOLOGY PROGRAM
NATIONAL INSTITUTES OF HEALTH

June 1, 1988

STANFORD UNIVERSITY SCHOOL OF MEDICINE
Edward H. Shortliffe, Principal Investigator
Edward A. Feigenbaum, Co-Principal Investigator

# DEPARTMENT OF HEALTH AND HUMAN SERVICES
# PUBLIC HEALTH SERVICE
# NATIONAL INSTITUTES OF HEALTH

## DIVISION OF RESEARCH RESOURCES
## BIOMEDICAL RESEARCH TECHNOLOGY PROGRAM

## ANNUAL PROGRESS REPORT
## PART I., TITLE PAGE

| | |
|---|---|
| 1. PHS GRANT NUMBER: | 5 P41 RR00785-15 |
| 2. TITLE OF GRANT: | SUMEX<br>Stanford University Medical<br>Experimental Computer Resource |
| 3. NAME OF RECIPIENT INSTITUTION: | Stanford University |
| 4. HEALTH PROFESSIONAL SCHOOL: | School of Medicine |
| 5. REPORTING PERIOD: | |
| 5a. FROM: | 08-01-87 |
| 5b. TO: | 07-31-88 |
| 6. PRINCIPAL INVESTIGATOR: | |
| 6a. NAME: | Edward H. Shortliffe, M.D., Ph.D. |
| 6b. TITLE: | Associate Professor of Medicine<br>and Computer Science |
| 6c. SIGNATURE: | Edward H Shortliffe |
| 7. DATE SIGNED: | July 5, 1988 |
| 8. TELEPHONE: | 415-725-3387 |

# Table of Contents

# List of Figures

# II. Description of Program Activities

This section corresponds to the predefined forms required by the Division of Research Resources to provide information about our resource activities for their computerized retrieval system. These forms have been submitted separately and are not reproduced here to avoid redundancy with the more extensive narrative information about our resource and progress provided in this report.

## II.A. Scientific Subprojects

Our core research and development activities are described starting on page 14, our training activities are summarized starting on page 97, and the progress of our collaborating projects is detailed starting on page 127.

## II.B. Books, Papers, and Abstracts

The list of recent publications for our core research and development work starts on page 83 and those for the collaborating projects are in the individual reports starting on page 127.

## II.C. Resource Summary Table

The details of resource usage, including a breakdown by the various subprojects, is given in the tables starting on page 100.

# III. Narrative Description

## III.A. Summary of Research Progress

## III.A.1. Resource Overview

This is an annual report for year 15 of the SUMEX-AIM resource (grant RR-00785), the second year of a 5-year renewal period to support further research on applications of artificial intelligence in biomedicine. For the technical and administrative reasons discussed in earlier reports, the SUMEX project now includes the continuation of work on the development and dissemination of medical consultation systems (ONCOCIN) that had been supported before 1986 as resource-related research under grant RR-01631. Progress on core ONCOCIN research is therefore now reported here as well.

These combined efforts represent an ambitious research program to:

- Continue our long-range core research efforts on knowledge-based systems, aimed at developing new concepts and methodologies needed for biomedical applications.
- Substantially extend ONCOCIN research on developing and disseminating clinical decision support systems.
- Develop the core systems technology to move the national SUMEX-AIM community from a dependence on the central SUMEX DEC 2060 to a fully distributed, workstation-based computing environment.
- Introduce these systems technologies into the SUMEX-AIM community with appropriate communications and managerial assistance to responsibly phase out the central resource and DEC 2060 mainframe in a manner that will support community efforts to become self-sustaining and to continue scientific interactions through fully distributed means.
- Maintain our aggressive efforts at training and dissemination to help exploit the research potential of this field.

### III.A.1.1. SUMEX-AIM as a Resource

**SUMEX and the AIM Community**

In the fifteen years since the SUMEX-AIM resource was established in late 1973, computing technology and biomedical artificial intelligence research have undergone a remarkable evolution and SUMEX has both influenced and responded to these changing technologies. It is widely recognized that our resource has fostered highly influential work in biomedical AI -- work from which much of the expert systems field emerged -- and that it has simultaneously helped define the technological base of applied AI research.

The focus of the SUMEX-AIM resource continues to emphasize research on artificial intelligence techniques that guide the design of computer programs that can help with the acquisition, representation, management, and utilization of the many forms of medical knowledge in diverse biomedical research and clinical care settings -- ranging from biomolecular structure determination and analysis, to molecular biology, to clinical decision support, to medical education. Nevertheless, we have long recognized that the ultimate impact of this work in biomedicine will be realized through its assimilation with the full range of methodologies of medical informatics, such as data bases, biostatistics, human-computer interfaces, complex instrument control, and modeling. From the start, SUMEX-AIM work has been grounded in real-world applications, like systems for the interpretation of mass spectral information about biomolecular structures, chemical synthesis, interpretation of x-ray diffraction data on crystals, cognitive modeling, infectious disease diagnosis and therapy, DNA sequence analysis, experiment planning and interpretation in molecular biology, and medical instruction. Our current work extends this emphasis in application domains such as oncology protocol management, clinical decision support, protein structure analysis, and data base information retrieval and analysis. All of these research efforts have demanded close collaborations with diverse parts of the biomedical research community and the integration of many computational methods from those domains with knowledge-based approaches. Even though in the beginning the "AI-in-medicine" community was quite small, it is perforce no longer limited and easily-defined, but rather is spreading and is inextricably linked with the many biomedical applications communities we have collaborated with over the years. Driven both by the on-going diffusion of AI and by the development of personal computer workstations that signal the practical decentralization of computing resources, we must develop new resource communication and distributed computing technologies that will continue to facilitate wider intra- and inter-community communication, collaboration, and sharing of biomedical information.

The SUMEX Project has demonstrated that it is possible to operate a computing research resource with a national charter and that the services providable over networks were those that facilitate the growth of AI-in-Medicine. SUMEX now has a reputation as a model national resource, pulling together the best available interactive computing technology, software, and computer communications in the service of a national scientific community. Planning groups for national facilities in cognitive science, computer science, and biomathematical modeling have discussed and studied the SUMEX model and new resources, like the recently instituted BIONET resource for molecular biologists, are closely patterned after the SUMEX example.

The projects SUMEX supports have generally required substantial computing resources with excellent interaction. Today, with the dramatic explosion of high-performance workstations that are more and more generally available, the need for a central source of raw computing cycles has significantly diminished. In place of being a distributor of CPU cycles, SUMEX has become a communications cross-roads and a source of AI and computer systems software and expertise.

SUMEX has demonstrated that a computer resource is a useful "linking mechanism" for bringing together electronically teams of experts from different disciplines who share a common problem focus. AI concepts and software are among the most complex products of computer science. Historically it has not been easy for scientists in other fields to gain access to and mastery of them. Yet the collaborative outreach and dissemination efforts of SUMEX have been able to bridge the gap in numerous cases. About 40 biomedical AI application projects have developed in our national community and have been supported directly by SUMEX computing resources over the years -- many more have benefitted indirectly through access to the software, information, and advice offered by the SUMEX resource.

The integration of AI ideas with other parts of medical informatics and their dissemination into biomedicine is happening largely because of the development in the 1970's and early 1980's of methods and tools for the application of AI concepts to difficult professional-level problem solving. Their impact was heightened because of the demonstration in various areas of medicine and other life sciences that these methods and tools really work. Here SUMEX has played a key role, so much so that it is regarded as "the home of applied AI."

SUMEX has been the home of such well-known AI systems as DENDRAL (chemical structure elucidation), MYCIN (infectious disease diagnosis and therapy), INTERNIST (differential diagnosis), ACT (human memory organization), MOLGEN/BIONET (tools for DNA sequence analysis and molecular biology experiment planning), ONCOCIN (cancer chemotherapy protocol advice), SECS (chemical synthesis), EMYCIN (rule-based expert system tool), and AGE (blackboard-based expert system tool). In the past four years, our community has published a dozen books that give a scholarly perspective on the scientific experiments we have been performing. These volumes, and other work done at SUMEX, have played a seminal role in structuring modern AI paradigms and methodology.

### The Future of SUMEX-AIM

Given this background, what is the future need and course for SUMEX as a resource -- especially in view of the on-going revolution in computer technology and costs and the emergence of powerful single-user workstations and local area networking? The answers remain clear.

#### *Basic Research on AI in Biomedicine*

At the deepest research level, despite our considerable success in working on medical and biological applications, the problems we can attack are still sharply limited. Our current ideas fall short in many ways against today's important health care and biomedical research problems brought on by the explosion in medical knowledge and for which AI should be of assistance. Just as the research work of the 70's and 80's in the SUMEX-AIM community fuels the current practical and commercial applications, our work of the late 80's will be the basis for the next decade's systems.

The report of the panel on medical informatics [3], convened late in 1985 by the National Library of Medicine to review and recommend twenty-year goals for the NLM, listed among its highest priority recommendations the need to greatly expand and aggressively pursue an interdisciplinary research program to develop computational methods for acquiring, representing, managing, and using biomedical knowledge of all sorts for health care and biomedical research. These are precisely the problems which the SUMEX-AIM community has been working on so successfully and which will require work well beyond the five year funding period we have requested. It is essential that this line of research in the SUMEX-AIM community, represented by our core AI research, the ONCOCIN research, and our collaborative research groups, be continued.

#### *The Changing Role of the Central Resource*

At the resource level, there are changing, but still growing, needs for computing resources for the active AIM research community to continue its work over the next five years. The workstations to which we directed our attention in 1980 have now demonstrated their practicality as research tools and, increasingly, as potential mechanisms for disseminating AI systems as cost-effective decision aids in clinical settings such as private offices. The era of highly centralized general machines for AI research is rapidly coming to an end and is being replaced by networks of distributed but heterogeneous single-user machines sharing common information resources and communication paths among members of the biomedical research community.

Most of our community groups have been able to take advantage of local computing facilities, with SUMEX-AIM providing a central cross-roads for communications and the sharing of programs and knowledge. In its core research and development role, SUMEX-AIM has its sights set on the hardware and software systems of the next decade. We expect major changes in the distributed computing environments that are just now emerging in order to make effective use of their power and to adapt them to the development and dissemination of biomedical AI systems for professional user communities. In its training role, SUMEX is a crucial resource for the education of badly needed new researchers and professionals to continue the development of the biomedical AI field. The "critical mass" of the existing physical SUMEX resource, its development staff, and its intellectual ties with the Stanford Knowledge Systems Laboratory (KSL -- see Appendix A for a summary of current KSL research activities), make this an ideal setting to integrate, experiment with, and export these methodologies for the rest of the AIM community.

We will continue our *experimental* approach to distributed systems, learning to build and exploit distributed networks of these machines and to build and manage graceful software for these systems. Since decentralization is central to our future, we must learn its technical characteristics.

### *Resource Sharing*

An equally important function of the SUMEX-AIM resource is an exploration of the use of computer communications as a means for interactions and sharing between geographically remote research groups engaged in biomedical computer science research and for the dissemination of AI technology. This facet of scientific interaction is becoming increasingly important with the explosion of complex information sources and the regional specialization of groups and facilities that might be shared by remote researchers [2, 1]. Another of the key recommendations of the NLM medical informatics planning panel [3] was that high-speed network communication links be established throughout the biomedical research community so that knowledge and information can be shared across diverse research groups and that the required interdisciplinary collaborations can take place. Recent efforts to establish a national NSFNet, largely to support the supercomputer projects funded by NSF but also to replace and upgrade part of the national research community linkage that the now aging ARPANET has supported, have made important progress. Still, these efforts do not encompass the broad range of biomedical research groups that need national network access and to date, the NIH has not played an aggressive role in the interagency Research Internet coordination efforts. We must work to build a stronger institutional support for a National Research Network.

SUMEX continues to be an important pathfinder to develop the technology and community interaction tools needed to expand community system and communication resources. Our community building effort is based upon the developing state of distributed computing and communications technology and we have therefore turned our core systems research to actively supporting the development of distributed computing and communications resources to facilitate collaborative project research and continued inter-group communications.

**Summary of Long-term Goals**

- Maintain the synergistic relationship between SUMEX core system development, core AI research, our experimental efforts at disseminating clinical decision-making aids, and new applications efforts.
- Continue to serve the national AIM research community, less and less as a source of raw computing cycles and more and more as a transfer point for new technologies important for community research and communication. We will also continue our coordinating role within the community through electronic media and periodic AIM workshops.
- Maintain our connections to ARPANET, TELENET, and our local Ethernet and assist other community members to establish similar links by example, by integrating and providing enabling software, and by offering advice and support within our resources.
- Focus new computing resource developments on more effective exploitation of distributed workstations through better communication and cooperative computing tools, using transparent digital networking schemes.
- Enhance the computing environments of workstations so that only minimal dependency on central, general-purpose computing hosts remains and these mainframe time-sharing systems can be phased out eventually. Remaining central resources will include servers for communications, community information resources, and special computing architectures (e.g., shared- or distributed-memory symbolic multiprocessors) justified by cost-effectiveness and unique functionality.
- Incrementally phase in, disseminate, and evaluate those aspects of the local distributed computing resource that are necessary for continuing national AIM community support within this distributed paradigm. This will ultimately point the way towards the distributed computing resource model that we believe will interlink this community well into the next decade.
- Gradually and responsibly phase out the existing DEC 2060 machine as effective distributed computing alternatives become widely available. We expect this to be possible sometime during the next year of the continuation resource.
- Continue the central staff and management structure, essentially unchanged in size and function during the five-year transition period, except for the merging of the core part of the ONCOCIN research with the SUMEX resource.

## III.A.1.2. Significance and Impact in Biomedicine

Artificial intelligence is the computer science of representations of symbolic knowledge and its use in symbolic inference and problem-solving processes. Projects in the SUMEX-AIM community are concerned in some way with the application of AI to biomedical research and the resource has given strong impetus and support to knowledge-based system research in biomedicine. For computer applications in medicine and biology, this research path is crucial. Medicine and biology are not presently mathematically-based sciences; unlike physics and engineering, they are seldom capable of exploiting the mathematical characteristics of computation. They are essentially inferential, not calculational, sciences. If the computer revolution is to affect biomedical scientists, computers will be used as inferential aids.

The growth in medical knowledge has far surpassed the ability of a single practitioner to master it all, and the computer's superior information processing capacity thereby offers a natural appeal. Furthermore, the reasoning processes of medical experts are poorly understood; attempts to model expert decision-making necessarily require a degree of introspection and a structured experimentation that may, in turn, improve the quality of the physician's own clinical decisions, making them more reproducible and defensible. New insights that result may also allow us more adequately to teach medical students and house staff the techniques for reaching good decisions, rather than merely to offer a collection of facts which they must independently learn to utilize coherently.

Perhaps the larger impact on medicine and biology will be the exposure and refinement of the hitherto largely private heuristic knowledge of the experts of the various fields studied. The ethic of science that calls for the public exposure and criticism of knowledge has traditionally been flawed for want of a methodology to evoke and give form to the heuristic knowledge of scientists. AI methodology is beginning to fill that need. Heuristic knowledge can be elicited, studied, critiqued by peers, and taught to students.

The importance of AI research and its applications is increasing in general, without regard for the specific areas of biomedical interest. AI is one of the principal fronts along which university computer science groups are expanding. The pressure from student career-line choices is great. Federal and industrial support for AI research and applications is vigorous, although support specifically for biomedical applications continues to be limited. All of the major computer manufacturers (e.g , IBM, DEC, TI, UNISYS, HP, and others) are using and marketing AI technology aggressively and many software companies are putting more and more products on the market. Many other parts of industry are also actively pursuing AI applications in their own contexts, including defense and aerospace companies, manufacturing companies, financial companies, and others.

Despite the limited research funding available, there is also an explosion of interest in medical AI. The American Association for Artificial Intelligence (AAAI), the principal scientific membership organization for the AI field, has 7000 members, several thousand of whom are members of the medical special interest group known as the AAAI-M. Speakers on medical AI are prominently featured at professional medical meetings, such as the American College of Pathology and American College of Physicians meetings; a decade ago, the words *artificial intelligence* were never heard at such conferences. And at medical computing meetings, such as the annual Symposium on Computer Applications in Medical Care (SCAMC) and the international MEDINFO conferences, the growing interest in AI and the rapid increase in papers on AI and expert systems are further testimony to the impact that the field is having.

AI is beginning to have a similar effect on medical education. Such diverse organizations as the National Library of Medicine, the American College of Physicians, the Association of American Medical Colleges, and the Medical Library Association have all called for sweeping changes in medical education, increased educational use of computing technology, enhanced research in medical computer science, and career development for people working at the interface between medicine and computing. They all cite evolving computing technology and (SUMEX-AIM) AI research as key motivators. At Stanford, we have vigorous special programs for student training and research in AI -- a graduate program in Medical Information

Sciences and the two-year Masters Degree in AI program. All of these have many more applicants than available slots. Demand for their graduates, in both academic and industrial settings, is so high that students typically begin to receive solicitations one or two years before completing their degrees.

## III.A.1.3. Summary of Current Resource Goals

The following outlines the specific objectives of the SUMEX-AIM resource during the current three-year award period begun in August 1986. It provides an overall research plan for the resource and provides the backdrop against which specific progress is reported. Note that these objectives cover only the resource nucleus; objectives for individual collaborating projects are discussed in their respective reports in Section IV. Specific aims are broken into five categories: 1) Technological Research and Development, 2) Collaborative Research, 3) Service and Resource Operations, 4) Training and Education, and 5) Dissemination.

*1) Technological Research and Development*

SUMEX funding and computational support for core research is complementary to similar funding from other agencies (including DARPA, NASA, NSF, NLM, private foundations, and industry) and contributes to the long-standing interdisciplinary effort at Stanford in basic AI research and expert system design. We expect this work to provide the underpinnings for increasingly effective consultative programs in medicine and for more practical adaptations of this work within emerging microelectronic technologies. Specific aims include:

- Basic research on AI techniques applicable to biomedical problems. Over the next term we will emphasize work on blackboard problem-solving frameworks and architectures, knowledge acquisition or learning, constraint satisfaction, and qualitative simulation.

- Investigate methodologies for disseminating application systems such as clinical decision-making advisors into user groups. This will include generalized systems for acquiring, representing and reasoning about complex treatment protocols such as are used in cancer chemotherapy and which might be used for clinical trials.

- Support community efforts to organize and generalize AI tools and architectures that have been developed in the context of individual application projects. This will include retrospective evaluations of systems like the AGE blackboard experiment and work on new systems such as BB1, CARE, EONCOCIN, EOPAL, Meta-ONYX, and architectures for concurrent symbolic computing. The objective is to evolve a body of software tools that can be used to more efficaciously build future knowledge-based systems and explore other biomedical AI applications.

- Develop more effective workstation systems to serve as the basis for research, biomedical application development, and dissemination. We seek to coordinate basic research, application work, and system development so that the AI software we develop for the next 5-10 years will be appropriate to the hardware and system software environments we expect to be practical by then. Our purchases of new hardware will be limited to experimentation with state-of-the-art workstations as they become available for our system developments.

*2) Collaborative Research*

- Encourage the exploration of new applications of AI to biomedical research and improve mechanisms for inter- and intra-group collaborations and communications. While AI is our defining theme, we may consider exceptional applications justified by some other unique feature of SUMEX-AIM essential for important biomedical research. We will continue to exploit community expertise and sharing in software development.

- Minimize administrative barriers to the community-oriented goals of SUMEX-AIM and direct our resources toward purely scientific goals. We will retain the current user funding arrangements for projects working on SUMEX facilities. User projects will fund their own manpower and local needs; actively contribute their special expertise to the SUMEX-AIM community; and receive an allocation of system resources under the control of the AIM management committees. We will begin charging "fees for service" to Stanford users as DRR support for the DEC 2060 is phased out. Fees to national users will be delayed as long as financially possible.

- Provide effective and geographically accessible communication facilities to the SUMEX-AIM community for remote collaborations, communications among distributed computing nodes, and experimental testing of AI programs. We will retain the current ARPANET and TELENET connections for at least the near term and will actively explore other advantageous connections to new communications networks and to dedicated links.

*3) Service and Resource Operations*

SUMEX-AIM does not have the computing or manpower capacity to provide routine service to the large community of mature projects that has developed over the years. Rather, their computing needs are better met by the appropriate development of their own computing resources when justified. Thus, SUMEX-AIM has the primary focus of assisting new start-up or pilot projects in biomedical AI applications in addition to its core research in the setting of a sizable number of collaborative projects. We do offer continuing support for projects through the lengthy process of obtaining funding to establish their own computing base.

*4) Training and Education*

- Provide documentation and assistance to interface users to resource facilities and systems.

- Exploit particular areas of expertise within the community for assisting in the development of pilot efforts in new application areas.

- Accept visitors in Stanford research groups within limits of manpower, space, and computing resources.

- Support the Medical Information Science and MS/AI student programs at Stanford to increase the number of research personnel available to work on biomedical AI applications.

- Support workshop activities including collaboration with other community groups on the AIM community workshop and with individual projects for more specialized workshops covering specific research, application, or system dissemination topics.

*5) Dissemination*

While collaborating projects are responsible for the development and dissemination of their own AI systems and results, the SUMEX resource will work to provide community-wide support for dissemination efforts in areas such as:

- Encourage, contribute to, and support the on-going export of software systems and tools within the AIM community and for commercial development.
- Assist in the production of video tapes and films depicting aspects of AIM community research.
- Promote the publication of books, review papers, and basic research articles on all aspects of SUMEX-AIM research.

## III.A.2. Details of Technical Progress

This section gives an overview of progress for the nucleus of the SUMEX-AIM resource. A more detailed discussion of our progress in specific areas and related plans for further work are presented beginning on Page 19. Objectives and progress for individual collaborating projects are discussed in their respective reports on Page 127. These collaborative projects collectively provide much of the scientific basis for SUMEX as a resource and our role in assisting them has been a continuation of that evolved in the past. Collaborating projects are autonomous in their management and provide their own manpower and expertise for the development and dissemination of their AI programs.

### III.A.2.1. Progress Highlights

In this section we summarize highlights of SUMEX-AIM resource activities over the past year (May 1987 - April 1988), focusing on the resource nucleus.

- We made excellent progress in **core AI research**. We have begun to explore the design and use of very large knowledge bases with the hypothesis that both the problems of brittleness and over-specialization in current knowledge-based systems can be addressed by constructing large, multi-use knowledge bases (LMKB). A LMKB would 1) encode domain knowledge in greater depth and breadth than required for any specific task, 2) encode knowledge that cuts across many domains of expertise, and 3) serve as a core repository of knowledge to be accessed by large numbers of specific applications.

  Research has also progressed on several fundamental issues of AI, including knowledge representation, blackboard frameworks, parallel symbolic computing architectures, and machine learning. Work continues in PROTEAN and BB1 on the explicit representation of control knowledge, on the representation of geometric problem-solving knowledge in PROTEAN and PEAKS, on the representation of diagnosis expertise and the integration of a numerical simulator of a model system with an expert system in ABLE, and on the flexible, rich representation of control knowledge to facilitate modeling of problem-solving at the strategic level as well as at the tactical level. We have continued to develop the BB1 blackboard architecture for systems that reason about (control, explain, and learn about) their own actions. The BB1 system is being applied in a new domain, the real time monitoring of patients in an intensive care unit (BBICU).

  The parallel architectures work has developed a CAD (Computer Aided Design) system for hierarchical, multiple-level specification of computer architectures (SIMPLE); a parameterized, multiprocessor array emulation defined in SIMPLE's specification languages and running on SIMPLE's simulator (CARE); a set of extensions to Lisp for studying expressed concurrency in functional programming, object-oriented, and shared-variable models of concurrent computation (LAMINA); and two alternative parallel blackboard frameworks for expressing application problems (POLIGON and CAGE). These have been applied to several signal understanding problems with promising problem- solving speed-up.

  The machine learning work has concentrated on explanation-based generalization and chunking work in the SOAR framework, inductive rule learning, "apprenticeship" learning, and tools for acquiring knowledge and debugging knowledge structures. Our research in machine learning has focused on several distinct problem domains including medicine (NEOMYCIN/HERACLES), physics (ABLE), and biochemistry (PROTEAN) in addition to domain-independent investigations.

Work has also continued on reasoning with uncertainty to find ways of combining formal and informal approximate reasoning methods. Specifically, this research seeks to (1) develop techniques for using knowledge about problem-solving tradeoffs to dynamically optimize system performance for the user, (2) construct efficient algorithms for probabilistic reasoning, and (3) investigate pragmatic techniques for the elicitation of knowledge from experts.

- We have made continued significant progress in the **core ONCOCIN research** to generalize the tools for clinical trial management from the initial cancer chemotherapy management application. A major accomplishment this year from our work on the examination of the structures of protocols across medical subspecialties other than cancer chemotherapy (e.g., hypertension and insulin diabetes treatment) was the creation of a generalized knowledge acquisition tool designed to encode descriptions of clinical trials. The system is called PROTEGE and produces as its output a *computer program* which is an OPAL-like clinical trial protocol definition system specifically tailored for an arbitrary clinical area such as hypertension. This OPAL-like system can then be used to create the knowledge base (e.g., for hypertension) that drives an ONCOCIN-like patient/protocol management system.

  We continued development of the OPAL system for graphical knowledge acquisition to facilitate protocol definition and knowledge base entry for the ONCOCIN oncology application area. We began to explore alternative platforms for developing OPAL-like systems such as HyperCard on the MAC II. We also tested the usefulness of a "generic protocol viewing tool" based on a relational data base storage mechanism. We performed several experiments aimed at developing a single underlying storage mechanism (termed a "cell") from which various interface systems including the Interviewer flowsheet, a generalized spreadsheet utility, and the OPAL schema entry flowchart interface could be derived.

  We have begun a project to explore the integration of speech-recognition technology into the interface to ONCOCIN. The project uses a commercially available continuous speech recognition product and a prototype ONCOCIN adaptation now permits users to navigate the graphical interface and enter clinical data using spoken commands. The development of this system exploits the significant experience we have gained in distributed computing since the phonetic device, initial parsing software, and the ONCOCIN system all reside on different pieces of hardware.

  We released a new version of our ONCOCIN object language which has proven to be the most stable and powerful version to date. We have also continued to examine the issues of disseminating the ONCOCIN system into actual clinical settings.

- We have made excellent progress on the **core system development** work targeted at supporting the distributed AIM community. We elaborated our systems development plans further at a site visit held in August 1987 in response to our request to the National Advisory Research Resources Council to restore the final 2 years of our grant award. Following a special study section review and reconsideration by the Council, our plan and the full 5-year grant award were approved. In line with the review group recommendations, we have moved to sharply focus our development resources on a small number of standardized hardware and software configurations.

After consideration of many requirements and alternative systems for AIM community computing needs for the coming years to replace and upgrade the services of the 2060, we have chosen Apple Macintosh II workstations as the general computing environment for researchers and staff, TI Explorer Lisp machines (including the microExplorer Macintosh coprocessor) as the near-term high-performance Lisp research environment, and a SUN-4 as the central system network server (wide- and local-area network interfaces, file services, printing services, etc.). The bulk of this hardware was purchased with DARPA research funds and we are now beginning the installation and integration process. We have concentrated our current systems efforts on getting basic capabilities operational, such as for text processing, filing/archiving, printing, graphics, office management, system building tools, information resource access, and distributed system operation and management) etc.

Development work for the Mac/Explorer/SUN environments has been limited because of manpower cuts necessitated by NIH cuts in our funding awards. Available resources have focussed on providing remote access between workstations, integrating a solid support of the TCP-IP network protocol, and building a distributed electronic mail system. We have significantly refined the prototype distributed EMail system developed for the Xerox D-machines last year (a number of people are using the system routinely to manage their mail) and are porting this to the Explorer and adapting it for the Mac II.

We have started a project based on the MacWorkstation software licensed from APPLE which is designed to allow the Macintosh to start up programs on mainframes or other workstations and receive graphic and text output to the MAC in a seamless fashion. We plan to write a Common Lisp window package (based on Common Windows) that uses MacWorkstation so we can connect (via Ethernet, AppleNet or RS-232C/modem) to any of our Common Lisp engines and run the same piece of code on any of them.

One of the key issues in selecting the systems for our distributed computing environment was the performance of Common Lisp and to help make this evaluation, we undertook an informal survey of the performance of two KSL AI software packages, SOAR and BB1, on a wide variety of machines. Within a factor of two of the best performance, a considerable range of workstations based on stock microprocessor chips as well as specially microprogrammed Lisp chips have comparable performance. Even though performance gaps between microprogrammed Lisp systems and stock workstation implementations are narrowing, there still remains a significant difference in the quality of the development environments. We have attempted to distill the key features of the Lisp machine environments that would be needed in stock machine implementations in order to make them attractive in a development setting.

We have installed an interface between our Ethernet environment and the TELENET network by which many AIM users gain access to SUMEX that allows full access to distributed SUMEX resources. We have also continued to maintain the DEC 2060 environment to stay current with operating system and other environmental upgrades.

- We have continued the **dissemination of SUMEX-AIM technology** through various media. We have reorganized the distribution system for our AI software tools (EMYCIN, AGE, and BB1) to academic, industrial, and federal research laboratories, in order to make it more efficient and require less research staff time. We have also continued to distribute the video tapes of some of our research projects including ONCOCIN, and an overview tape of Knowledge Systems Laboratory work to outside groups. Our group has continued to publish actively on the results of our research, including more than 45 research papers per year in the AI literature and a dozen books in the past 5 years on various aspects of SUMEX-AIM AI research. We assisted and participated actively in the AIM Workshop sponsored by AAAI and held at Stanford (Ramesh Patil from MIT was the Program Chairman) and hosted a number of AIM community visitors at our Stanford research laboratory this past year.

- The **Medical Information Sciences program**, begun at Stanford in 1983 under Professor Shortliffe as Director, has continued its strong development over the past year. The specialized curriculum offered by the MIS program focuses on the development of a new generation of researchers able to support the development of improved computer-based solutions to biomedical needs. The feasibility of this program resulted in large part from the prior work and research computing environment provided by the SUMEX-AIM resource. It has recently received enthusiastic endorsement from the Stanford Faculty Senate for an additional five years, has been awarded renewed post-doctoral training support from the National Library of Medicine with high praise for the training and contributions of the SUMEX-AIM environment from the reviewing study section, and has received additional industrial and foundation grants for student support. This past year, MIS students have published many papers, including several that have won conference awards.

- We have continued to recruit **new user projects and collaborators** to explore further biomedical areas for applying AI. A number of these projects are built around the communications network facilities we have assembled, bringing together medical and computer science collaborators from remote institutions and making their research programs available to still other remote users. At the same time we have encouraged older mature projects to build their own computing environments thereby facilitating the transition to a distributed AIM community. A substantial number of projects have moved to their own computing resources, including SOAR, under Dr. Paul Rosenbloom at USC/ISI in Los Angeles; the Logic Group projects (DART, Intelligent Agents, and MRS) under Professor Michael Genesereth at Stanford; Hierarchical Models of Human Cognition (CLIPR), under Professors Walter Kintsch and Peter Polson at the University of Colorado; Problem Solving Expertise (SOLVER), under Professors Paul Johnson and William Thompson at the University of Minnesota; RXDX, under Professor Robert Lindsay at the University of Michigan; and Computer-Based Exercises in Pathophysiologic Diagnosis, under Professor J. Robert Beck at Dartmouth College.

SUMEX user projects have made good progress in developing and disseminating effective consultative computer programs for biomedical research. These systems provide expertise in areas like cancer chemotherapy protocol management, clinical diagnosis and decision-making, and molecular biology. We have worked hard to meet their needs and are grateful for their expressed appreciation (see Section IV).

## III.A.2.2. Core ONCOCIN Research

ONCOCIN is a data management and therapy advising program for complex cancer chemotherapy experiments. The development of the system began in 1979, following the successful generalization of MYCIN into the EMYCIN expert system shell. The ONCOCIN project has evolved over the last eight years: the original version of ONCOCIN ran on the time-shared DEC computers using a standard terminal for the time-oriented display of patient data. The current version uses compact workstations running on the Ethernet network with a large bit-mapped displays for presentation of patient data. The project has also expanded in scope. There are three major research components: 1) ONCOCIN, the therapy planning program and its graphical interface; 2) OPAL, a graphical knowledge entry system for ONCOCIN; and 3) ONYX, a strategic planning program designed to give advice in complex therapy situations. Each of these research components has been split into two parts: continued development of the cancer therapy versions of the system, and generalization of each of the components for use in other areas of medicine. This annual report will describe our work on each of the components: implementation of ONCOCIN workstations in the Stanford clinic, knowledge acquisition research, and research to generalize ONCOCIN for application in clinical trial domains other than medical oncology (E-ONCOCIN).

A major highlight of this year was the creation of a generalized knowledge acquisition tool designed to encode descriptions of clinical trials. The system, named PROTEGE, was the Ph.D. thesis work of Mark Musen. The output of PROTEGE is an OPAL-like input system designed for one clinical area such as hypertension. This input system (HTN-OPAL) can then be used to create the hypertension knowledge base for an E-ONCOCIN like system. This experiment was carried out this year for both the hypertension and oncology domains. Details of this project are described later in this report.

### 1 - Overview of the ONCOCIN Therapy Planning System

ONCOCIN is an advanced expert system for clinical oncology. It is designed for use after a diagnosis has been reached, focusing instead on assisting with the management of cancer patients who are receiving chemotherapy. Because anticancer agents tend to be highly toxic, and because their tumor-killing effects are routinely accompanied by damage to normal cells, the rules for monitoring and adjusting treatment in response to a given patient's course over time tend to be complex and difficult to memorize. ONCOCIN integrates a temporal record of a patient's ongoing treatment with an underlying knowledge base of treatment protocols and rules for adjusting dosage, delaying treatment, aborting cycles, ordering special tests, and similar management details. The program uses such knowledge to help physicians with decisions regarding the management of specific patients.

A major lesson of past work in clinical computing has been the need to develop methods for integrating a system smoothly into the patient-care environment for which it is intended. In the case of ONCOCIN, the goal has been to provide expert consultative advice as a by-product of the patient data management process, thereby avoiding the need for physicians to go out of their way to obtain advice. It is intended that oncologists use ONCOCIN routinely for recording and reviewing patient data on the computer's screen, regardless of whether they feel they need decision-making assistance. This process replaces the conventional recording of data on a paper flowsheet and thus seeks to avoid being perceived as an additive task. In accordance with its knowledge of the patient's chemotherapy protocol, ONCOCIN then provides assistance by suggesting appropriate therapy at the time that the day's treatment is to be recorded on the flowsheet. Physicians maintain control of the decision, however, and can override the computer's recommendation if they wish. ONCOCIN also indicates the appropriate interval until the patient's next treatment and reminds the physician of radiologic and laboratory studies required by the treatment protocol.

### 2 - Implementation of the ONCOCIN Workstation in the Stanford Clinic

In mid-1986, we placed the workstation version of ONCOCIN into the Oncology Day Care clinic. This version is a completely different program from the version of ONCOCIN that was available in the clinic from 1981-1985-- using protocols entered through the OPAL program, with a new graphical data entry interface, and revised knowledge representation and reasoning component. One person in the clinic (Andy Zelenetz) became primarily responsible for making sure that our design goals for this version of ONCOCIN were met. His suggestions included the addition of key protocols and the ability to have the program be useful for clinicians as a data management tool if the complete treatment protocol had not yet been entered into the system. Additional fellows are being trained on a very stable release of ONCOCIN that became available in early 1988. A version of the system was sent to the University of Pittsburgh for evaluation. In addition, ONCOCIN will become available for presentation at the National Library of Medicine Artificial Intelligence Demonstration Center. For these various efforts, Janice Rohn has created an extensive user manual, sample patient interactions, and reminder cards to shorten the training period for ONCOCIN.

The process of entering a large number of treatment protocols in a short period of time led to other research topics including: design of an automated system for producing meaningful test cases for each knowledge, modification of the design of the time-oriented database and the methods for accessing the database, and the development of methods for graphically viewing multiple protocols that are combined into one large knowledge base. These research efforts will continue into the next year. In addition, some of the treatment regimens developed for the original mainframe version are still in use and can be transferred to the new version of ONCOCIN.

We also received new insights about the design of the internal structures of the knowledge base (e.g., the relationship between the way we refer to chemotherapies, drugs, and treatment visits). We will continue to optimize the question-asking procedure, the method for traversing the plan structure in the knowledge base, and consider alternative arrangements used to represent the structure of chemotherapy plans. Although we have concentrated our review of the ONCOCIN design primarily on the data provided by additional protocols, we know that non-cancer therapy problems raise similar issues. The E-ONCOCIN effort is designed to produce a domain-independent therapy planning system that includes the lessons learned from our oncology research.

### 3 - E-ONCOCIN: Domain Independent Therapy Planning

During the past two years, our E-ONCOCIN research has concentrated on understanding how protocols in medicine vary across subspecialties. We are examining several application areas: the intensive care unit, insulin treatment for diabetes, hypertension protocols, and both standard and complex cancer treatment problems. The diagnosis and therapy selection for patients in the intensive care unit is a natural application area because it is based on changing data and the need to determine the response to therapy interventions. In addition, it is an area where reasonable mathematical models of the respiratory system can be integrated into the expert system. We also felt that the area of insulin treatment for diabetes would be a good area to explore. Like cancer chemotherapy, the treatments for diabetes continues over a long period of time and has been the area of intensive protocol development. Unlike cancer chemotherapy, the treatment plan must handle multiple treatments in one day and deemphasizes the use of multiple drugs (although there are a variety of types of insulin). During 1987, using the medical literature and several internists in the medical computer science research group (Mark Frisse, Mark Musen, and Michael Kahn), we performed knowledge acquisition experiments for insulin treatment of diabetes. The proposed structure for the knowledge base was implemented using the object-oriented programming language upon which ONCOCIN has been based. These experiments, like those of adding more protocols to ONCOCIN, demonstrated the need for changes in the way that the knowledge base can access the time-oriented data base that records patient data and previous conclusions. The relationships between the different doses and types of insulin treatments will also require alternative ways of building treatment hierarchies. Thus, our initial experiments have shown that many of the elements of the ONCOCIN design are sufficiently general for other application areas, but that some specific elements (particularly the representation of temporal events) will have to be generalized. A description of our revised temporal representations will appear in a forthcoming thesis by Michael Kahn of University of California at San Francisco. During the coming year, we will continue our knowledge acquisition experiments and design a version of the E-ONCOCIN system that is separate from the ongoing "clinic version."

### 4 - OPAL: Graphical Knowledge Acquisition Interface

OPAL is a graphical environment for use by an oncologist who wishes to enter a new chemotherapy protocol for use by ONCOCIN or to edit an existing protocol. Although the system is designed for use by oncologists who have been trained in its use, it does not require an understanding of the internal representations or reasoning strategies used by ONCOCIN. The sysrem may be used in two interactive modes, depending on the type of knowledge to be entered. The first permits the entry of a graphical description of the overall flow of the therapy process. The oncologist manipulates boxes on the screen that stand for various steps in the protocol. The resulting diagram is then translated by OPAL into computer code for use by ONCOCIN. Thus, by drawing a flow chart that describes the protocol schematically, the physician is effectively programming the computer to carry out the procedure appropriately when ONCOCIN is later used to guide the management of a patient enrolled in that protocol.

OPAL's second interactive mode permits the oncologist to describe the details of the individual events specified in the graphical description. For example, the rules for administering a given chemotherapy will vary greatly depending upon the patient's response to earlier doses, intercurrent illnesses and toxicities, hematologic status, etc. For example, one form permits the entry of an attenuation schedule for an agent based upon the patient's white count and platelet count at the time of treatment. Tables such as this are generally found in the written version of chemotherapy protocols. Thus OPAL permits oncologists to enter information using familiar forms displayed on the computer's screen. The contents of such forms are subsequently translated into rules and other knowledge structures for use by ONCOCIN.

### *4.1 - Status of the OPAL System*

The OPAL is one of the few graphical knowledge acquisition systems ever designed for expert systems. Even fewer are designed to be used as the main method for entering knowledge as opposed to a proof of concept implementation. We have pursued three directions in the development of the OPAL system, also in response to the large number of protocols entered through this system during the last year. The first direction is the modification of graphical forms needed to allow the entry of facts that did not show up in the protocols used to test the initial version of OPAL. OPAL continues to assume that most of the knowledge to be entered will have very stereotyped forms, e.g., dose attenuations for most treatment toxicities are based on a comparison of only one laboratory measurement at a time, such as using the BUN to adjust for renal toxicity. We sometimes need much more complex ways of stating the scenarios in which dose adjustments may be necessary. This need has led us in a second direction, towards a "lower-level" rule entry approaching the syntax of the reasoning component of ONCOCIN, but using graphical input devices where applicable. A major accomplishment of this last year was to experimentally combine the OPAL and ONCOCIN programs into one working program, and to completely enter knowledge from OPAL using both the high level tools and lower level rule editors, but without needing to make changes at the ONCOCIN side of the system. Work on graphical replacements for low level rule entry is the master's project of Eric Sherman.

The OPAL program maps the information provided on the graphical forms into a complex data structure (called the IDS) that represents the required knowledge to specify the contents of a protocol. This data structure is used for copying information from one protocol to another, and as the basis for the creation of the ONCOCIN knowledge base. Our experiments with OPAL, and our intention to generalize OPAL use outside of oncology protocols, suggests that we reorganize the OPAL program to use a relational database to store its knowledge. We have patterned the database after an existing database query syntax. Because no databases exist for the InterLisp language upon which OPAL is based, we reimplemented the database from its written description. The database structure is completed, and was the basis for the PROTEGE knowledge acquisition experiments.

With changes in the future of dedicated lisp processors, we began to explore alternative platforms for developing OPAL-like systems. We have begun experiments using HyperCard on the MAC II. We also tested the usefulness of a "generic protocol viewing tool" based on a relational-database storage mechanism, in preparation for a OPAL design based on relational DBMS technology. We also performed several experiments aimed at developing a single underlying storage mechanism (termed a "cell") from which various interface systems including the Interviewer flowsheet, a generalized spreadsheet utility, and the OPAL schema entry flowchart interface could be derived.

### **5 - Generalized Knowledge Acquisition through PROTEGE**

Mark Musen designed and implemented the first version of the PROTEGE knowledge-acquisition-system development tool. PROTEGE is used to collect information which describes the concepts (both entities and their relationships) in an application area for which a skeletal-planning type of expert system would be useful, concentrating on clinical trials. The system acquires the "ontology" of a domain through a series of fill-in-the-blank forms and a "flowchart-entry" tool. These concepts are then mapped onto a set of generic forms, which, in turn , create a knowledge acquisition tool for the application area.

PROTEGE makes use of the forms management system built for the original OPAL, and a newly developed relational database management system written for the Xerox InterLisp-D workstations. The output of PROTEGE is an OPAL-like set of forms tailored to the special structures of the application area. To test these ideas, we first reimplemented portions of the OPAL interface from a high level description of oncology. After the translation process to the ONCOCIN reasoning program, a consultation was run that matched the manually built system. This experiment was then repeated for the area of hypertension protocols for which ONCOCIN had never been specifically designed. With some minor generalizations to the oncocin reasoner and interviewer, we were able to run a hypertension consultation.

## 6 - Speech Input to Expert Systems

### *6.1 - Prototype Speech Hardware/Software System*

In 1987 we began a project to explore the integration of speech-recognition technology into the interface to ONCOCIN running on the XEROX lisp workstations. The project uses a commercially available continuous-speech-recognition product loaned by the vendor, Speech Systems, Inc. (SSI) of Tarzana, California. The speech recognizer consists of a custom processor, called the Phonetic Engine(R) and a suite of software modules called the Phonetic Decoder(TM). The Phonetic Decoder(TM) is running on a SUN 3/75.

The development of this project requires significant experience in distributed computing since the phonetic device, initial parsing software and the ONCOCIN system all reside on different pieces of hardware. One of the early steps is to allow the Lisp machine to remotely control the parsing software on the SUN. We built an interpreter for communicating between the speech software library running on the SUN and the Xerox Lisp machine. This interpreter reads lisp-style function calls corresponding to the speech library routines and returns lisp-style results as remote procedure call (RPC) mechanism. We then wrote the library of corresponding Lisp stub routines and a function to connect to the SUN workstation (through a programmatic TELNET) and start the server. In normal operation, we call the 'C'-based speech library routines from Lisp as if they were Lisp functions. We did a number of updates to both components of the server program (on the SUN and Lisp machine) as various SSI updates came out, including a major revision when SSI doubled the number of library routines, greatly increasing the server's capabilities.

Eventually we plan to replace the programmatic TELNET and custom RPC language with a new version of the system that uses the SUN RPC language. This will include the Xerox Lisp interface to the SUN RPC package developed by SUMEX; we were not able to use this interface initially as it did not exist early enough and required a release of Xerox Lisp that supports Common Lisp which the ONCOCIN system was not yet using. This eventual change will allow more efficient communications between the machines, allowing us to move larger data structures much faster. This will also tie in to the proposed method of communications between Lisp and non-Lisp routines that Xerox plans to use when their system migrates to the SUN workstations. The Lisp routines that correspond to the various SSI library routines will not appear to change to the application programs so they should not require any modification when the underlying RPC mechanism is changed.

We created a prototype system that permits users to navigate the graphical interface and enter clinical data using speech. The system uses the location of the cursor on the screen to provide a context for choosing candidate grammars with which to attempt recognition of a user's utterance. The system dynamically re-orders the list of candidate recognition grammars based on the dialog history. Albeit with limitations on the legal grammars, it is now possible to carry on most of the ONCOCIN data acquisition steps using speech alone or speech plus pointing with the mouse. In addition, some elements such as the neural toxicities can be entered as textual descriptions and automatically encoded as one the 1-4 point scale used on flowsheet forms.

In order to translate an utterance back into an action that can occur in the ONCOCIN interface, we need the ability to reparse the text string returned by the SSI equipment. The SSI equipment uses (potentially complex) syntax, built up of various classifications, to understand sentences but returns just the ASCII component of the actual sentence; you can not get it in terms of the original classifications in the syntax (which are generally semantically significant). To overcome this problem, we devised a whole new syntax format based on Lisp and our OZONE object language in which one devises a grammar from which the SSI syntax files are generated. Then when the ASCII string is returned, the original syntax object is used to parse the string into a parse tree that relates directly to the grammar definition. We can now process the returned information at a much higher level than was possible with the simple ASCII text. In addition, the semantic elements of the parse can be added to syntactic structures used to encode the sentence.

The latest additions to this part of the system include exhaustive and random sentence generation as Lisp data structures (previously only partially available just on the SUN workstation) as well as word list generation from the syntax objects. These features generate both in-core representations as well as file representations which will potentially free us from any dependency on SSI's equipment (versus another manufacture's speech input device).

We wrote graphics software that plots the sentence returned by the speech equipment along with the changes in amplitude, pitch, accuracy scores and other information. This helps us to understand how SSI's 'black box' operates, particularly in situations where it fails. In addition, we wrote programs to make it possible to return all the best syntactic matches from the speech equipment. The current SSI software can either return the one best answer or all the possible word mappings (unfortunately regardless of syntax). We believe the latter would be useful when the most likely parse is associated with a low reliability score. We could then look at all the highest scoring candidates and evaluate what they do and do not have in common for creating a clarifying query to the speaker. The candidate sentences are composed from the word mapping data which is immense and needs to be compressed, filtered and manipulated to be of any use (the parser/generator is used as a filter to remove the non-syntactic word mappings). Unfortunately, though it runs faster with each improvement, this doesn't look reasonable for real time use; currently a three word utterance takes about two to four minutes to process into a candidate list of half a dozen possible sentences. It gives us an indication of what would be useful data structures to process on the SUN (to improve speed and lower data transmission).

### *6.2 - Speech Experiments*

We are performing experiments to (1) enhance the system's grammars with a wider range of phrases clinicians actually use when talking to a computer and (2) gain insights into clinicians' models of spoken interaction with advice systems so that we may ground our interface design in observed practice.

In order to assess how physicians would speak to a computer in an ideal situation without constraints or prior assumptions, we are conducting a series of experiments which simulate continuous-speech understanding by computers. The setting of these experiments includes a hidden computer operator simulating the output of ONCOCIN if it had the ability to understand the spoken input, as well as a video camera to record both audio and visual clues. Typed responses from the operator are translated back as actions on the computer display as well as audio responses through the use of a speech synthesizer. It appears to the subject as if the computer is understanding and responding to their speech. The physicians use ONCOCIN in the same manner as it is used in the clinic when they see patients, but with the added capability of speech input. These experiments enable us to both build up a basic vocabulary for the speech system as well as examine subtle linguistic issues to guide future directions.

One component of these experiments was the use of speech synthesis as an output medium. Using an inexpensive board based on the General Instrument ASCII-to-phoneme and phoneme-to-voice chip set, we built a driver for the Xerox Lisp machines. This driver included an application software transparent misspelling dictionary facility to correct for inaccuracies in the speech board.

### 7 - Object Language Support for ONCOCIN Project

We released a new version of our object language at the start of this past year which has proven to be the most stable and powerful version to date. There have been a number of minor bug fixes and several feature additions over the course of the year but for the most part the system as required much less attention than in previous years. The number of new systems being built on it (like our speech work) continues to increase. Future planning for the system consists of determining whether or not it should be converted to Common Lisp, based on whether object systems available under Common Lisp are sufficient for our needs, and if we do convert it what it would it look like if properly integrated with that language.

### 8 - Personnel

Samson Tu has been primarily responsible for the design of E-ONCOCIN, Michael Kahn has developed the temporal representations used by the system, Clifford Wulfman has been involved with extensions the the data entry interface and the extensions to the interface in order to add speech input. Samson and Cliff were responsible for extensions to their programs to support the PROTEGE effort. David Combs has been involved with the knowledge acquisition interface and provided major programming support for the PROTEGE effort. Janice Rohn has been involved with the entry of protocols, interaction with physicians using the system, documentation of the system, and execution of the speech experiments. Ellen Isaacs, a Ph.D. student in Psycholinguistics has helped to design the speech experiments. Christopher Lane has developed the object-oriented systems software upon which the entire ONCOCIN system is designed and the systems software and parsing programs used in the speech project.

# III.A.2.3. Core AI Research

### 1 - Rationale

Artificial Intelligence (AI) methods are particularly appropriate for aiding in the management and application of knowledge because they apply to information represented symbolically, as well as numerically, and to reasoning with judgmental rules as well as logical ones. They have been focused on medical and biological problems for well over a decade with considerable success. This is because, of all the computing methods known, AI methods are the only ones that deal explicitly with symbolic information and problem solving and with knowledge that is heuristic (experiential) as well as factual.

Expert systems are one important class of applications of AI to complex problems -- in medicine, science, engineering, and elsewhere. An expert system is one whose performance level rivals that of an human expert because it has extensive domain knowledge (usually derived from an human expert); it can reason about its knowledge to solve difficult problems in the domain; it can explain its line of reasoning much as an human expert can; and it is flexible enough to incorporate new knowledge without reprogramming. Expert Systems draw on the current stock of ideas in AI, for example, about representing and using knowledge. They are adequate for capturing problem-solving expertise for many bounded problem areas. Numerous high-performance, expert systems have resulted from this work in such diverse fields as analytical chemistry, medical diagnosis, cancer chemotherapy management, VLSI design, machine fault diagnosis, and molecular biology. Some of these programs rival human experts in solving problems in particular domains and some are being adapted for commercial use. Other projects have developed generalized software tools for representing and utilizing knowledge (e.g., EMYCIN, UNITS, AGE, MRS, BB1, and GLisp) as well as comprehensive publications such as the three-volume *Handbook of Artificial Intelligence* and books summarizing lessons learned in the DENDRAL and MYCIN research projects.

There is considerable power in the current stock of techniques, as exemplified by the rate of transfer of ideas from the research laboratory to commercial practice. But we also believe that today's technology needs to be augmented to deal with the complexity of medical information processing.

Our core research goals, as outlined in the next section, are to analyze the limitations of current techniques and to investigate the nature of methods for overcoming them. Long-term success of computer-based aids in medicine and biology depend on improving the programming methods available for representing and using domain knowledge. That knowledge is inherently complex: it contains mixtures of symbolic and numeric facts and relations, many of them uncertain; it contains knowledge at different levels of abstraction and in seemingly inconsistent frameworks; and it links examples and exception clauses with rules of thumb as well as with theoretical principles. Current techniques have been successful only insofar as they severely limit this complexity. As the applications become more far-reaching, computer programs will have to deal more effectively with richer expressions and much more voluminous amounts of knowledge.

Expert systems are being developed that impact nearly every field of human endeavor: medicine, manufacturing, financial services, diagnosis of machinery, geology, molecular biology and structural design, to name a few. Each new instance is a confirmation of the hypothesis that knowledge is power. In each system, expert level problem-solving performance is obtained by using relatively simple and uniform reasoning methods which access an extensive body of domain knowledge. The ability of these systems lies not in their superior reasoning capabilities but in the *specific* concepts, facts, methods, models, etc. that can be brought to bear on the problem. The *knowledge is power* hypothesis has received so much confirmation that we now assert it as the *knowledge principle.* A corollary to the Knowledge Principle is that significant improvements in the power of knowledge-based systems will be derived primarily from the ability to access large amounts of knowledge.

During the past year we have begun to explore the design and use of very large knowledge bases. In the last twenty years we have learned how to build intelligent programs that perform at a high level of competence on specialized tasks within narrowly defined domains. These programs traditionally access small to modest-sized knowledge bases specialized to the prescribed task. In contrast, we have started on a long-range research effort that will result in a *large, multi-use knowledge base* (LMKB).

We believe construction of a LMKB is an essential step toward resolving two fundamental problems plaguing the current generation of expert systems. The first is *brittleness*: current systems can exhibit only a very narrow range of expert behavior, and their performance falls off precipitously at the limits of their expertise. The second problem is *over-specialization*: a knowledge base constructed to support of one type of expert task (e.g., diagnosis) cannot be used to support other types of tasks (e.g., design).

Our hypothesis is that both the problems of brittleness and over-specialization can be addressed by constructing large, multi-use knowledge bases. A LMKB would 1) encode domain knowledge in greater depth and breadth than required for any specific task, 2) encode knowledge that cuts across many domains of expertise, and 3) serve as a core repository of knowledge to be accessed by large numbers of specific applications.

This report documents progress on the basic or core research activities within the Knowledge Systems Laboratory (KSL), funded in part under the SUMEX resource as well as by other federal and industrial sources. This work explores a broad range of basic research ideas in many application settings, all of which contribute in the long term to improved knowledge based systems in biomedicine.

## 2 - Highlights of Progress

In the last year, research has progressed on several fundamental issues of AI. As in the past, our research methodology is experimental; we believe it is most fruitful at this stage of AI research to raise questions, examine issues, and test hypotheses in the context of specific problems, such as management of patients with Hodgkin's disease. Thus, within the KSL we build systems that implement our ideas for answering (or shedding some light on) fundamental questions; we experiment with those systems to determine the strengths and limits of the ideas; we redesign and test more; we attempt to generalize the ideas from the domain of implementation to other domains; and we publish details of the experiments. Many of these specific problem domains are medical or biological. In this way we believe the KSL has made substantial contributions to core research problems of interest not just to the AIM community but to AI in general.

Progress is reported below under each of the major topics of our work. Citations are to KSL technical reports listed in the publications section.

### *2.1 - Knowledge Representation*

How can the knowledge necessary for complex problem solving be represented for its most effective use in automatic inference processes? Often, the knowledge obtained from experts is heuristic knowledge, gained from many years of experience. How can this knowledge, with its inherent vagueness and uncertainty, be represented and applied?

Work continues in PROTEAN and BB1, with its explicit representation of control knowledge (see the summary of Blackboard Architectures below). In particular, we have advanced our methods for representation of geometric problem solving knowledge in PROTEAN and PEAKS (see PROTEAN section of this report.) We have developed an application in a new domain of diagnosis and correction of errors in a linear accelerator beam line, the ABLE project. In this we have explored issues of representation of diagnosis expertise, and have developed a method that incorporates a numerical simulator of a model system with an expert system (see discussion under Knowledge Acquisition and Learning below.) In addition, we continued research on NEOMYCIN which has a component for using a flexible, rich representation of control knowledge to facilitate modeling of problem solving at the strategic level as well as at the tactical level.

[See KSL technical reports KSL-87-58 and KSL-87-62.]

### *2.2 - Blackboard Architectures and Control*

How can we design flexible control structures for powerful problem solving programs? How can we use these structures effectively in many problem domains? How can we represent processes and reason about their behavior, and perform intelligent actions under *real-time* requirements?

We have continued to develop the BB1 blackboard architecture for systems that reason about -- control, explain, and learn about -- their own actions. In the last year, we have significantly extended the system-building support and run-time capabilities of the BB1 system. These extensions include (a) declarative representation of large bodies of factual and heuristic knowledge; (b) integration of multiple reasoning skills in a single system; (c) dynamic control under real-time constraints. We have also implemented the following application-independent components: (a) declarative representation for device structure, function, faults, and repairs; (b) reasoning modules for associative and model-based diagnosis; and (c) an asynchronous communications interface.

During the past year, we began work on an advisory system, called BB-ICU (see also the separate collaborative project report on Page 129), to support patient monitoring in a surgical intensive-care unit (SICU). Briefly, intensive care patients are critically ill individuals who require life-support devices, such as respirators or dialysis machines, to perform some of their vital functions. During their stay in the intensive care unit, patients are monitored closely and gradually weaned from life-support devices in coordination with their changing physiological status and other therapeutic interventions.

We began by visiting the SICU at the Palo Alto VAMC to observe monitoring procedures and operations. We worked with Dr. Adam Seiver to enumerate and characterize component intensive-care monitoring tasks and to delineate the space of relevant knowledge. We developed an ontology and representation scheme for important categories of knowledge (e.g., anatomy, physiology, pathology, therapy) and assessed our approach by implementing a small amount of knowledge in each category. We then enumerated key architectural requirements for BB-ICU and identified those not met in existing AI architectures.

During the fall of 1987, we elaborated our initial ideas in the context of more focused design and implementation activities. Exploiting and extending our BB1 architecture [KSL-84-16, KSL-88-22], we developed: (a) an asynchronous I/O subsystem to provide integrated and asynchronous perception, action, and cognition; (b) an intelligent I/O mediator to translate, interpret, and filter low-level data on behalf of the application system; and (c) an I/O manager to coordinate the mediator's activities with an application system's dynamic attentional focus. Working within our BB* knowledge representation environment [KSL 86-38], we implemented representations of the anatomy, physiology, and pathology of the respiratory system as instances of corresponding elements of a generic flow system. We developed reasoning components for continuous data interpretation and associative diagnosis of observed symptoms. We also developed reasoning components that instantiate generic models, such as the flow system model, to explain the causal relations underlying associative diagnoses or to hypothesize plausible diagnoses in the absence of associative knowledge. We demonstrated the application of the reasoning components and respiratory knowledge to interpret, diagnose, and explain respiratory data of the sort continuously monitored in the intensive-care unit. BB-ICU Demo-1 comprises independently implemented versions of each of these system components. BB-ICU Demo-2, which we completed in April, integrates these system components.

Some of this work is reported in recent technical reports [KSL 87-31, KSL 87-67, KSL 88-20, KSL 88-22). Other reports are in preparation. In addition, we have given talks describing this work at: the Carnegie-Mellon University Symposium on Architectures for Intelligence, Boeing Computer Services in Seattle, Wa., Advanced Decision Systems in Mountain View, Ca., and the DARPA Planning Workshop in Austin, Tx.

### *2.3 - Advanced Architectures*

Many applications, such as process planning and control, maintenance, troubleshooting, environmental control, and crisis management require knowledge-based systems that can cope with large amounts of data and that produce responses in real-time. The current hardware and software architectures for knowledge-based systems cannot support such requirements. The most promising approach for achieving orders of magnitude improvement in the quantitative performance of knowledge-based systems is by exploiting concurrency on multiprocessor systems. Based on near-term projections for integrated circuit technologies, it is clear that highly parallel multiprocessor computers consisting of 100's to 1000's of processors and realizing a variety of concurrent architectures can be built. The major issue is whether such computers can be effectively used to enhance the performance of knowledge-based systems. Since 1985, the Knowledge Systems Laboratory at Stanford University has been investigating this issue.

The goals and technical approach of this project, largely supported by DARPA under the Strategic Computing Program, have been discussed in previous annual reports. To summarize briefly, we seek to achieve two to three orders of magnitude speedup in the execution of knowledge-based systems, by identifying and exploiting sources of concurrency at all levels of system design: the application level, the problem solving framework level, the programming language level and the hardware systems architecture level. Due to the inherent complexity of the task and the lack of theoretical foundations for parallel computation with ill-structured problems, we have taken an empirical approach. During the first phase of the project, which was concluded in July 1987, we made specific choices at each of the system levels, i.e. taken a "vertical slice" through the design space, and have conducted several experiments to investigate the effects of a wide variety of parameters on performance.

Our research methodology is:

- Select specific knowledge-based system applications, primarily signal understanding applications.
- Encode these applications following various proposed concurrent software models.
- Evaluate the qualitative and quantitative performance of the applications running on simulated multiprocessor machines with respect to varying hardware parameters, for example, number of processors and communication protocols, and varying software organizations, for example, degree of control centralization.

In the following discussion, we present the major components of our project, and for each component we describe its current status.

### 2.3.1 - SIMPLE/CARE Multiprocessor Simulation System

Simulation of systems at an architectural level can offer an effective way to study critical design choices if (1) the performance of the simulator is adequate to examine designs executing significant code bodies -- not just toy problems or small application fragments, (2) the details of the simulation include the critical details of the design, (3) the view of the design presented by the simulator instrumentation leads to useful insights on potential problems with the design, and (4) there is enough flexibility in the simulation system so that the asking of unplanned questions is not suppressed by the weight of the mechanics involved in making changes either in the design or its measurement.

SIMPLE/CARE is a simulation system which satisfies these requirements. It forms the foundation for our empirical investigations of software architectures and hardware system architectures for concurrent knowledge-based systems. SIMPLE is a CAD (Computer Aided Design) system for hierarchical, multiple level specification of computer architectures and includes an associated mixed-mode, event-based simulator. CARE is a parameterized, multiprocessor array emulation specified in SIMPLE's specification languages and running on SIMPLE's simulator. Our simulation system is in use by several research groups at Stanford, and it has been ported to several external sites including NASA Ames Research Center. A tutorial was held in January, attended by representatives from the DoD, NASA and Boeing, which described the CARE/SIMPLE system, as well as the LAMINA programming interface (see below). The attendees received instruction in use of the system for making measurements of the performance of various simulated multiprocessor applications.

The SIMPLE/CARE system is currently implemented in ZetaLisp and executes on Texas Instrument Explorer and Symbolics 3600-class Lisp workstations. We have recently started a reimplementation of the system in Common Lisp. One of our proposed objectives during the coming year is to complete this reimplementation. A Common Lisp version of SIMPLE/CARE will make it portable to a wide variety of computer systems including Sun and MicroVAX workstations. This development will necessarily be an ongoing task as Common Lisp standards, in particular, window standards, evolve and as the inevitable commercial reinterpretations of standards emerge.

The SIMPLE design specification system has design operators for automatically generating array type multiprocessor architectures from a "unit cell" specification. There is considerable interest, both at Stanford and elsewhere, in using the system to specify and simulate other types of multiprocessor architectures. A second proposed objective is to augment SIMPLE's design operators with recursive operators for the generation of architectures using, for example, hierarchical busses or recursive interconnection nets such as Omega nets.

### 2.3.2 - LAMINA Programming Interface

LAMINA provides extensions to Lisp for studying expressed concurrency in functional programming, object oriented, and shared variable models of concurrent computation. The implementation of the support for all three computational models is based on the common notion of a stream, a data type which can be used to express pipelined operations by representing the promise of a (potentially infinite) sequence of values. LAMINA also provides system support for the management of software pipelines and dynamic structure creation, relocation, and reclamation in a multiprocessor, multi-address-space system.

Algorithms and applications written in LAMINA may be run on the SIMPLE/CARE simulation system in order to study their execution on alternative multiprocessor architectures.

The development of LAMINA was essentially completed during the past year, and the software is now reasonable stable. In order to make LAMINA available in the community, we intend to port it to Common Lisp. We also expect that the application research will motivate various extensions to the LAMINA programming interface.

### 2.3.3 - Poligon Problem Solving Framework

Poligon [KSL 86-19, KSL 88-04] is a framework for the development of Blackboard-like applications on a (simulated) multiprocessor. It consists of:

- A compiler, which compiles a high-level description of the Blackboard's structure and the Knowledge to be applied by the system, to run on a distributed memory multiprocessor.
- A run-time system which provides a debugging and testing environment for Poligon programs as well as run-time support.

Both the compiler and the run-time system are thoroughly integrated with the program development environment of TI Lisp machines, the machine on which the execution of Poligon programs are simulated.

Serial Blackboard Systems are implemented with the Nodes being represented as records on the Blackboard. The Knowledge is encoded in Knowledge Sources. These are typically compiled into procedures which are invoked by the Blackboard System's kernel. There is some form of scheduler for the Knowledge, which invokes one Knowledge Source after another. The Blackboard and the Knowledge Base both share the same address space, though they are functionally distinct. Knowledge Sources are "invoked" (executed) as a result of changes in the Blackboard placing that change event in a queue used by the scheduler. The scheduler repeatedly picks a Knowledge Source which is interested in the type of event at the end of the queue.

Experiments with Poligon are by no means complete, but we have learned a number of lessons thus far. Some of these lessons are enumerated below.

- It is very hard to write any program which implements either a framework, such as Poligon or an application such as those which have been mounted on Poligon. This is due largely to asynchronous side effects. A system with better formal properties would be less error prone in this respect but might well make much less efficient use of the hardware. These difficulties could also be caused by an insufficiency of mechanisms to control coherency in Poligon.

- In order to produce a reliable program it is necessary to write code which makes no assumptions about anything that any other part of the system might be doing. Failure to do so results in brittle systems.

- In order to achieve a coherent solution it was found to be necessary to develop a number of programming methodologies. For example, the creation of blackboard Nodes is tricky. Because each element is likely to represent some real-world object, it is important either to provide a mechanism for resolving the conflict caused by multiple asynchronous requests to create an element that represents the same thing or to provide a mechanism for managing the creation of Nodes. Poligon opts for the latter approach.

### 2.3.4 - CAGE Problem Solving Framework

CAGE [KSL 86-41, KSL 88-02] is a framework for building and executing applications as a concurrent blackboard system. CAGE is based on the AGE [KSL 80-29] serial blackboard framework. It includes mechanisms for the concurrent execution of knowledge sources, rules and parts of rules. The CAGE user has complete control over which of these mechanisms are used. CAGE is designed to execute on a shared-memory, multiprocessor system with tens to hundreds of processors. It is implemented using Qlisp, a concurrent dialect of Lisp designed for multiprocessors with a single, shared address space. CAGE currently executes on a shared-memory variant of CARE simulated using the SIMPLE simulation system.

We are nearing completion of a series of end-to-end experiments for evaluating the utility and performance of the CAGE concurrent blackboard framework. During the coming year we intend to complete these experiments and disseminate the results.

### 2.3.5 - CAGE, Poligon and LAMINA Comparative Experiments

During the past two years we have been developing application software and machine architecture models to support a series of end-to-end experiments comparing various concurrent programming systems for knowledge-based applications. The goals of these experiments are to:

- Obtain quantitative comparisons of the performance of the programming systems.

- Gain insights into how different concurrent programming models lead to different (or similar) application decomposition and organization.

- Force the refinement of the concurrent programming systems so as to better support application development.

- Gain insights into the ease or difficulty of writing application code in each of the programming systems.

The common application for these experiments is Elint [KSL 86-69], a real-time, knowledge-based system for integrating pre-processed, passively acquired radar emissions from aircraft. This Elint application has been implemented in three different concurrent programming systems: LAMINA, Poligon and CAGE.

Each of the implemented applications are executed and evaluated using various input data sets and varying numbers of processors.

Application code written in either LAMINA or Poligon compiles to code which executes on the CARE architecture. CAGE, however, is targeted toward a single address space, shared variable multiprocessor architecture. CAGE is implemented in QLisp, a concurrent Lisp for shared variable multiprocessors. To support CAGE we had to develop a multiprocessor "blackboard machine" variant of CARE. This blackboard machine models a shared variable architecture and includes the mechanisms and instruments necessary to manage and study memory contention. The architecture implements the blackboard and the control data structures in global, shared memory. It directly supports the CAGE system and application code written in QLisp.

During the past year we have:

- Completed the implementation of the the Elint application in each of the three concurrent programming systems.
- Completed the development of the blackboard machine variant of CARE.
- Developed an experiment plan for the comparative studies.
- Developed a new measure of speedup as a function of the number of processors in a multiprocessor system. This measure is useful for evaluating system performance of real time applications and is based on the concept of maximum sustainable input data rate.
- Completed the first set of experiments for each of the three programming systems.

### 2.3.6 - The AIRTRAC Application

AIRTRAC [KSL 86-20] is the primary application driving our development of concurrent knowledge-based system programming methodologies. Also, it is one of the basic applications used for our multiprocessor architecture performance experiments. AIRTRAC is a knowledge-based signal interpretation and information fusion system. The system attempts to identify, track, and predict the future behavior of aircraft. In particular, it attempts to recognize aircraft which might be engaged in covert activity, for example, smuggling. The inputs to AIRTRAC are periodic radar tracking system reports, a priori, filed flight plans for some aircraft, and occasional intelligence reports about suspected covert activity.

AIRTRAC is designed to be sufficiently complex and realistic to adequately test various ideas about concurrent problem solving on multiprocessor machine architectures. The AIRTRAC application involves continuous input data streams, typical of real-time signal interpretation problems. Such problems often require a level of computational power two to three orders of magnitude beyond what is currently available. Moreover, the application uses data-driven, expectation-driven and model-driven styles of reasoning. These reasoning styles encompass a wide range of paradigms in artificial intelligence.

The AIRTRAC Data Association Module and associated experiments were completed as of summer, 1987 [KSL 87-34]. The experiments were performed using the SIMPLE/CARE multiprocessor simulation system. They demonstrated that almost linear speedup as a function of the number of processors can be achieved (at least up to 100 processors) for a periodic data-driven knowledge-based system such as the Data Association Module.

During the past year, the design and knowledge acquisition for the Path Association Module was completed. Over one half of the LAMINA code for this module has been implemented and debugged.

The completed AIRTRAC application will provide an end-to-end example of a concurrent, knowledge-based signal interpretation system. It will demonstrate the benefits and costs of implementing and executing such systems on multiprocessor architectures. Also, the application is sufficiently complex that it will serve as important test case for evaluating multiprocessor architectures for knowledge-based systems and "tuning" the engineering parameters for such systems.

### *2.4 - Knowledge Acquisition and Machine Learning*

Our research in machine learning has focused on several distinct problem domains including medical (NEOMYCIN/HERACLES), physics (ABLE), and biochemical (PROTEAN) in addition to domain-independent investigations. We also are motivated by the need for effective tools for knowledge acquisition and maintenance of knowledge bases (IMPULSE and STROBE for FRM, BBEDIT, KSEDIT with BB1).

#### 2.4.1 - Learning by Chunking

*Chunking* is a learning mechanism that acquires rules from goal-based experience. SOAR is a general problem-solving architecture with a rule-based memory that can use the learning capabilities of chunking for the acquisition and use of macro-operators. Rosenbloom et al. are investigating chunking in SOAR and find that chunking obtains extra scope and generality from its intimate connection with the sophisticated problem solver (SOAR) and the memory organization of the production system. Another emphasis in SOAR is Explanation-Based Learning, a powerful technique that generalizes concepts learned from examples. In this past year, SOAR has demonstrated acquisition of diagnostic problem solving knowledge (similar to that in NEOMYCIN), learning from multiple examples and from analogy, and learning attribute-value information to acquire features of a single object incrementally, reusing known objects as values of attributes. The work on SOAR is continuing under Dr. Paul Rosenbloom at ISI in Los Angeles.

#### 2.4.2 - Inductive Rule Learning

In previous reports, we discussed the work of Buchanan, et al. on incremental learning process from examples with the rule-learning system RL (described in the 1986 SUMEX report). This work has continued, and has been applied to the domain of linear accelerator physics (see below). Results from the RL research indicate that intelligent selection of instances based upon knowledge of the state of the evolving theory results in a faster convergence of an evolving theory toward the target concept, requiring many fewer cases for learning.

In the paper Simulation-Assisted Inductive Learning, to be presented at the 1988 AAAI conference in Minnesota, Buchanan, Clearwater, et al. describe the work done as a collaborative effort between the Rule Learner project and the Automated Beam

Line Experiment project. The focus of this work is to show how RL can be used in a real-world domain with sparse data by working with a simulator which numerically models the domain. The domain is the classification and location of faults in a particle accelerator. This study demonstrates the effectiveness of RL even in this noisy, numerical domain. Some problems in generating the best examples from which to learn rules and how to learn the best rules from a given set of examples are explored. In addition, methods of weighing the evidence when several rules fire are being investigated.

### 2.4.3 - Learning Apprentice

In addition, we continued several investigations of methods for bui'ding knowledge bases for knowledge-based programs. Knowledge engineering remains the "standard" method of building a knowledge base for commercial systems, so we have investigated ways of making that more efficient. Technical reports KSL 87-58 and KSL 87-62 describe some of the results, pertaining mostly to the principles of starting with a sound problem solving strategy and of exploratory programming. Reports KSL 87-60 and KSL 87-67 describe work on learning apprentice methods.

In last year's SUMEX report, we reported results on the ODYSSEUS apprenticeship learning program, described by Wilkins in KSL 86-63, which is designed to refine and debug knowledge bases for the HERACLES expert system shell. ODYSSEUS analyzes the behavior of a human specialist using two underlying domain theories, a *strategy theory* for the problem solving method (heuristic classification), and an *inductive theory* based on past problem solving sessions. ODYSSEUS improves the knowledge base for the expert system shell, identifying bugs in the system's knowledge in the process of following the line-of-reasoning of an expert, serving as a knowledge acquisition subsystem. The system can also be used as part of an intelligent tutor, identifying problems in a novice's understanding and serving as student modeler for tutoring systems.

Wilkins, et al. illustrate that an explicit representation of the problem solving method and underlying theories of the problem domain provide a powerful basis for automating learning for expert system shells [KSL 86-62]. In the last year we began the creation of a case library from medical records for the NEOMYCIN domain. This library is essential for apprenticeship learning in ODYSSEUS in three ways. First, experiments have shown that the existing knowledge bases are too impoverished to follow the reasoning of an expert, and a case library will allow an induction system to automatically expand the domain knowledge to overcome this limitation. Second, when the learning system fails to explain an action of a student or expert correctly, the critic component of the system generates thousands of conjectures. To filter these requires a case library. Finally, the case library will allow us to demonstrate that apprenticeship learning can improve the performance of an expert system.

### *2.5 - Pragmatic Approaches to Reasoning Under Uncertainty*

The goal of this project is to investigate pragmatic approaches to computer-based probabilistic reasoning systems. In the past, artificial intelligence researchers have often avoided probability theory for reasoning with uncertainty because of the perception that the application of probability is invariably associated with a commitment to intractable algorithms and an inordinate amount of knowledge acquisition time. The development of efficient probability inference and assessment techniques will allow investigators to apply a theoretically justified theory of belief entailment to complex problems. Specifically, this project seeks to (1) develop techniques for using knowledge about problem-solving tradeoffs to dynamically optimize the value of computer performance to the user, (2) construct efficient algorithms for probabilistic reasoning, and (3) investigate pragmatic techniques for the elicitation of knowledge from experts.

### 2.5.1 - Reasoning about inference tradeoffs

Research on reasoning tradeoffs has focused on the time vs. quality-of-result tradeoff in several domains. Additionally, tradeoffs arising with the representation of knowledge and explanation of inference within a probabilistic framework have been identified [KSL 88-13] The growing perspective of this research is that problems traditionally ascribed to knowledge representation, inference, and explanation in probability-based reasoning systems have been encountered because of insufficient attention given to tradeoffs under bounded or varying resources available for engineering, computation, or cognition [KSL 88-13,87-28].

Notable research on inference tradeoffs during this past year has been the implementation of a prototype strategic reasoner for the analysis of tradeoffs. Fundamental issues of control under varying resource limitations were explored with simple sorting algorithms [KSL 88-3]. Some of this work will be reported in an article appearing in the AAAI conference this summer. Other work on inference tradeoffs has focused on applying a similar computational architecture to control the selection of alternative belief-network inference strategies [KSL 87-64].

Work has been carried out on the implementation and characterization of useful classes of probabilistic approximation algorithms that contain explicit tradeoffs. There has been special interest in the development of flexible strategies for reasoning under uncertainty with varying resource limitations. Recent work along these lines has focused on the development of modified versions of a probabilistic inference technique developed by Pearl [KSL 88-27]. The new approach allows the performance of the algorithm to be "gracefully degraded" under resource limitations, through pruning the consideration of terms in accordance with their expected effect on the final answer. Preliminary empirical analysis of the time/accuracy tradeoff for this algorithm has been carried out on a multiply-connected belief network.

### 2.5.2 - Efficient probabilistic inference algorithms

During the past year, two of our primary goals have been to implement several known probabilistic inference algorithms and begin to test their efficiency. In particular, we have implemented Pearl's algorithm for multiply-connected belief networks and in the process we have gained important insights into the nature of the algorithm [KSL 88-27]. These insights have allowed us to make design choices that yield an efficient implementation the algorithm. In particular, we have designed and developed a fast method for finding and using a cutset of nodes for inference in networks that contain complex loops. We also are working to improve the efficiency of updating belief networks that do not contain loops; our current algorithm is able to update a singly-connected network of propositional variables at a rate of approximately 50 nodes per second on a Macintosh II in Turbo Pascal.

Our implementation of the Pearl algorithm has been successfully tested using a set of benchmarks of varying complexity that we have developed. These benchmarks soon will be used to test comparatively the computational time complexity of several additional inference algorithms. We also have begun a collaboration with a researchers at the University of Aalborg in Denmark who are leading an effort in Europe to develop expert systems based on belief networks. We have already shared some of our benchmarks with them. In the coming months we anticipate increased collaboration that will include comparison of benchmark timing results and the exchange of algorithms.

The Pearl algorithm that we have implemented will serve as one type of exact algorithm. In addition, we are currently implementing a probabilistic inference algorithm by Lauritzen and Spiegelhalter. The Pearl and Lauritzen/Spiegelhalter algorithms will be our initial set of exact algorithms. We also plan to study approximation and heuristic techniques for probabilistic inference. This is particularly important in light of our proof that exact probabilistic inference using belief networks is NP-hard [KSL 87-27]. We previously implemented a stochastic algorithm and will use this as our initial approximation algorithm. An neural-network algorithm will be used as an initial heuristic algorithm. We anticipate that by the end of the summer we can begin to compare these algorithms on sets of theoretical and real inference problems. These tests will provide important information for designing new algorithms in the coming year.

Several other researchers in the Medical Computer Science Group are currently using our inference algorithms to develop expert systems based on belief networks. Systems that presently are being developed include 1) an intelligent anesthesia monitor, 2) a diagnostic system for the Intensive Care Unit, and 3) a system that assists in evaluating the statistical validity of a clinical drug trial report. These applications have given us practical feedback about the level of inference performance that is necessary in real domains. Although these applications are still in the early stages of development, they suggest that improved inference efficiency will be a critical issue in producing practical expert systems that are based on belief networks.

During the past year we have developed a general knowledge engineering environment called KNET (Knowledge NETwork) on a Macintosh II in MPW Pascal [see Chavez article]. KNET is a flexible graphical interface system for entering a belief network [see Lehmann article] and running cases using a belief network. It provides a general software system foundation from which to experiment with different methods for pragmatic probabilistic reasoning. For example, a key feature is that KNET provides a modular environment in which different inference techniques can be tested. We recently were able quickly to install Pearl's inference algorithm into KNET. We soon plan to have several other inference algorithms running in KNET.

Although the emphasis in the early stages of this work has been on developing an initial set of algorithms, we have also designed several new techniques. One technique uses dynamic programming to solve efficiently a large class of complex probabilistic inference problems. We have not yet implemented and tested this algorithm. However, it appears that for some types of complex belief network topologies this algorithm will be very fast relative to current techniques. Another method that was recently developed in our group allows any belief network algorithm to be used to solve decision problems (i.e., influence diagram problems) [KSL 88-28]. This general method significantly broadens the scope of application of our work on belief network algorithms. Of particular interest, it allows currently available exact, approximation, and heuristic belief network algorithms to be easily adapted to solve decision making problems.

### 2.5.3 - Probability assessment

During the past year, the majority of the research outlined for the reasoning-by-two assessment method, now called *similarity networks*, was completed. We began by implementing the similarity network approach in Turbo Pascal on the Macintosh II. We then evaluated the knowledge assessment tool by using the program to build a small component of the Pathfinder knowledge base, a module that helps a novice pathologist classify spherical structures seen in a lymph-node tissue section. Using the program, our expert was quickly able to identify the morphologic features relevant to the classification task and was able to specify the dependencies among these features.

Upon using the program for probability assessment, it became clear that a generalization of the similarity network would be useful for reducing the number of assessments required. In particular, it became clear that arbitrary sets of hypotheses should be allowed to be clustered together and labeled as "similar." A pen and pencil approach to eliciting probabilities in this manner was developed and used to assess the probabilities required for the entire lymph-node pathology domain. In assessing the probabilities, it was assumed that all observations are conditionally independent on each hypothesis. The approach was quite successful as it reduced the number of probability assessments required of the expert by a factor of approximately twenty (from 30,000 assessments to roughly 1,500).

Finally, the performance of the knowledge base constructed in this manner was evaluated [KSL 88-38]. In doing so, a new evaluation metric based in decision-theory was developed. The results of the evaluation demonstrated that the performance of the knowledge base was close to that of the expert. However, the results also showed that there is still room for improving the knowledge base through the representation of dependencies among features.

### 2.5.4 - Collaborations

During the past year we have continued to maintain contact with a number of Stanford faculty who are interested in the research goals of this grant. We have also collaborated with visitors from outside of Stanford. In particular, Prof. Max Henrion visited our group during the Autumn quarter. We continue to communicate frequently with him about our common research interests. We also have benefited from visits by Dr. David Spiegelhalter (a statistician from the Medical Research Council in England), Dr. Stig Andersen (a computer scientist from Aalborg University in Denmark), and Prof. Ross Shachter (a Stanford faculty member on sabbatical at Duke University).

Most of the members of our group were able to attend the recent conference on Influence Diagrams in Berkeley. We have submitted several papers to the AAAI Uncertainty Workshop this year and most of us plan to attend this workshop in August.

## III.A.2.4. Core System Development

### 1 - Introduction

In this section we describe progress on our core system development and work toward a distributed AIM community. In last year's report, we discussed the motivations and plans for our core system development work along four dimensions: 1) the motivation for the shift of the SUMEX-AIM community from a central mainframe-based computing resource model to a largely distributed workstation-based model; 2) the prospects for workstation technology and vendor support for a diverse distributed AIM community; 3) the projected core SUMEX-AIM systems tasks needed to complement vendor developments to realize distributed community operation; and 4) the integration, dissemination, and management of the shift of the AIM community from a centralized to a more distributed operation, including the remaining central resource functions. These were expanded still further at a site visit held in August 1987 in response to our request to the National Advisory Research Resources Council to restore the final 2 years of our grant award. Following a special study section review and reconsideration by the Council, our plan and the full 5-year grant award were approved. The review group's concluding recommendations included the following guidelines:

> "Consistent with its charter as a national resource, SUMEX should focus its systems activities on producing a distributed medical research environment that can be easily reproduced at other sites. It should also continue to play the important role it plays today as a repository of systems information and expertise for the medical AI research community, as well as the larger computer science community. However, it should avoid trying to be all things to all people and should focus its attention on a small number of standardized hardware and software configurations. A strong effort should be made to acquire information about related systems activities at other sites and to avoid duplication of effort. These guidelines should be used to establish priorities among the proposed set of system activities and to apply effort appropriately."

The review committee's recommendation was very much in line with our own goals to more sharply focus our development resources and much of our effort over the past year has been devoted to that end. Since our 1980 renewal proposal in which our move to distributed workstation technologies began, we had taken on the development and support of a wide array of systems including mainframes (DEC 2060 and 2020), network servers (2 DEC VAX 11/750 UNIX file servers, a SUN 3/180 UNIX file server, a Xerox file server, 7 network laser printers, Ethernet gateways and TIP's, and ARPANET and TELENET wide-area access systems), and workstations (50 Xerox D-machines, 20 TI Explorers, 6 Symbolics machines, 5 Hewlett-Packard 9836 workstations, and 3 SUN 3/75 workstations). Whereas we cannot drop support for these systems irresponsibly, we resolved to pick a much more limited environment on which to focus our long-term systems development efforts and to phase out support for the other systems as quickly as possible.

In summary, we have chosen Apple Macintosh II workstations as the general computing environment for researchers and staff, TI Explorer Lisp machines (including the microExplorer Macintosh coprocessor) as the near-term high-performance Lisp research environment, and a SUN-4 as the central system network server (wide- and local-area network interfaces, file services, printing services, etc.).

## 2 - Distributed System Evaluation and Selection

### *2.1 - Design Goals*

In planning for AIM community computing needs for the next few years to replace and upgrade the powerful and easy-to-use general computing tools and network services of the 2060, several goals were identified:

- The work environment should be modern and combine graphics, pointing, and traditional keyboard modalities of interaction, as it is expected to be the primary work environment for some years to come.
- The system should support the most powerful AI research and Lisp development environment available today, possibly involving special-purpose hardware.
- The system should support small-to-medium-size AI and Lisp-based research work without requiring special hardware.
- The cost per person should be low enough as to permit putting a machine on or near every desk and to consider the system as a potential AI delivery environment.
- The system should integrate well into a heterogeneous computing environment typical of AIM research work.
- The system should be capable of editing, organizing, and printing large documents, such as theses and books.
- The system should be capable of generating and editing state-of the art graphics.
- The design should be incrementally extendable and augmentable as new hardware and software technologies appear and as the number of users fluctuates.
- The design should be simple enough as to refocus our systems work on a smaller number of machines and cost-effective enough as to be replicable at smaller AIM sites that wish to benefit from our experience.
- The design should permit easy data sharing and exchange with collaborators at other sites and within Stanford University.

In addition to user-related computing tools, other heavily-used network services traditionally provided by the shared 2060, must be replaced. These include wide-area network access (ARPANET and TELENET), electronic mail (transmission/routing, reception, and user access), community bboards, file service, and print spooling.

### *2.2 - Evaluation Results*

We examined many potential configurations before deciding on the solution involving Mac II's, microExplorer's, and the SUN-4 . Many of the considerations were technical in terms of the tools and services provided by the systems and many had to do with user preferences for interface style and environment.

Timesharing machines were eliminated for their lack of modern interactive productivity tools. (The many reasons for the trend from timesharing to workstations have been discussed in previous annual reports. The pressures behind this trend have grown stronger with time.)

Xerox lisp machines were eliminated by virtue of their uniqueness and the questionable future of the hardware product line. Stand-alone TI (and other similar) Lisp machines were eliminated by virtue of their uniqueness and high cost and lack of general computing tools for mail, document preparation, etc.

Sun workstations were eliminated by virtue of their relatively higher cost and engineering orientation and their dependence on the UNIX user interface which received almost uniformly negative comment in a KSL user questionnaire about computing environment preferences.

IBM PC's were eliminated because of current limitations in their primitive operating system, window system, and interface style, when compared with the Macintosh.

Of course, this kind of evaluation is very complex and the above reflects only a summary of key issues. There are many reasons for or against any of the above machines not fully enumerated here. In the end, we chose the Apple Macintosh II primarily because of the following considerations:

- the Mac has a powerful, intuitive, and consistently applied icon-based user interface that facilitates wide use and effectiveness.
- The Mac II is a powerful machine (Motorola 68020-based) with an open architecture that provides long-term configuration flexibility (e.g., for color, coprocessing units like the microExplorer, memory, i/o devices, etc.).
- The Macintosh is popular and is used by a growing number of our collaborators. Many members of the AIM community specifically endorsed the Macintosh as their machine of choice and many already have Macintoshes at home.
- Apple gives educational institutions a substantial discount. This and the low price of third-party disks meant we could put a Mac on almost every desk.
- There is a large variety of third-party hardware and software available. Competition and volume mean lower prices; especially when compared to Sun or DEC third-party offerings.
- Texas Instruments announced the *microExplorer*, a board-level product which gives the user a custom VLSI lisp machine inside his Mac, for a fraction of the cost of TI's stand-alone *Explorer* workstation product.

To replace the network service functions of the SUMEX-AIM 2060, we chose a Sun 4/280 system. We investigated competitive systems, for example, the DEC VAX and Pyramid Technology 9000 series product line. Configurations are available from 3.5 to 25 MIPS and with individual I/O channel speeds up to 11 MBytes/Sec. However, we decided that the SUN-4 was more cost effective, had more popularity at Stanford and at other universities, and offered outstanding research network service software. This configuration is at the beginning of its product cycle and can be expected to serve for many years to come.

The SUN 4/280 approach was made even more attractive by the opportunity to have it equipped with state-of-the-art 900 megabyte disks. This option was presented to us after a review of second-source disks indicated that SUN's then offered Fujitsu Double Eagle units (575 megabytes) were not the optimum in cost-effectiveness.

### *2.3 - Configuration*

The working plan we eventually settled on called for a Sun 4/280, 69 Macintosh II workstations, 10 TI microExplorer upgrades, 7 Kinetics FastPath gateways, and a combination of upgrade packages yielding 2 20 page/minute PostScript laser printers.

The Sun 4/280 is configured with 32 megabytes of memory and 1.8 gigabytes of disk space. By choosing this package we were able to purchase the system for 40% off list price. An ARPANET interface for the Sun server is available and will be purchased in the near future. This will make the new server more readily available to AIM users outside Stanford. This server is currently up and undergoing test and configuration.

By a special arrangement, we purchased 66 Macintosh II computers directly from Apple, each with 1 MB of memory (no disk, no display). At this writing, almost all of the the Macintoshes have been installed. Three units have proven defective and are being repaired under warranty.

A package of 10 microExplorer upgrades was ordered from Texas Instruments. They arrived during the preparation of this report and will be installed shortly. (The features of the microExplorer are described in the *Explorers* section of this report.)

10 100 MB disks were purchased from *Rodime* and installed in the Macintoshes receiving the microExplorer upgrades. (A large paging disk is required by the microExplorer's use of virtual memory.) The balance of the Macintoshes were outfitted with 20 MB disks, also purchased from Rodime. The choice of Rodime disks was suggested by their low price (30% less than Apple's higher-education discount price) and long warranty (12 months). All disks have been installed and so far none have failed. Small disks were deemed sufficient, as users are encouraged to keep their files on the Sun file server (using the AppleShare filing protocol) for reasons of data backup and security.

Our experience with large displays on other workstations suggested that we wanted the largest displays the market could offer at a reasonable price. We chose 33 *Moniterm* 19" displays and 33 *Moniterm* 24" displays, which we were able to obtain in a package at 40% off list price. Our experience with other third-party Macintosh displays told us that a resolution of no greater than 72 dots/inch is easiest on the eyes. Both of the models we purchased conform to this resolution.

Also from our previous experience with Macintoshes, we knew that many applications require more than 1 MB of memory. In our initial purchase, we specified 12 4 MB memory upgrades. These were installed in 12 Macintoshes used primarily by staff and student developers of Mac software. The original 1 MB of memory was removed from each of these machines and added to 12 other machines, making a total of 2 MB in each of those machines. (10 of the 12 were the microExplorers.) We have already concluded that our memory demands require that we do the same with the remaining 1 MB Mac's. Apple's memory orders are now backlogged 5 months, so we have ordered from National Semiconductor instead.

To network the majority of the Macintoshes in the near term, we chose the *Farallon PhoneNet* system which enabled us to reuse terminal wiring we had previously installed in all offices and student areas. In addition to this reason, we chose PhoneNet over Apple's *LocalTalk* wiring system because PhoneNet permits nets 3-4 times larger (by reason of different shielding and impedance characteristics).

To connect the PhoneNet networks to the SUMEX-AIM Ethernet, we chose to install 7 *Kinetics FastPath* gateways. The FastPath is a commercial spin off resulting from the SUMEX work on the *SEAGATE* gateway. Owing to an earlier royalty payment agreement with Kinetics, we were able to procure the FastPath gateways at no cost. The number of gateways was chosen primarily because of the limited throughput characteristics of PhoneNet (230.4 Kb), but also to provide for hardware redundancy.

To provide printing services for this number of Macintoshes, it was necessary to procure additional PostScript printers. Although Apple offers a substantial discount on its LaserWriter products to its higher-education customers, the 8 page per minute maximum speed (typically less) was deemed too low for our demand printing needs. (Other complaints about the design of the LaserWriter concerned the small paper tray and toner capacity.) As noted in the *Printing Services* section of this report, we obtained a no-cost PostScript upgrade for our 20 page per minute *Imagen 3320* printer in consideration of our having beta tested the upgrade product. Our experience with the basic 3320 product over the past year has been positive. Our positive reaction to the PostScript upgrade convinced us that duplicating the configuration in our other offices was a good idea. We were able to do this by upgrading an Imagen 12/300 to a 3320 PostScript product at a 30% discount. The 3320 is appealing mostly for its print quality, speed, and minimal maintenance requirements. It holds a ream of paper and can print on 11" x 17" paper.

### *2.4 - More Details about the Transition Plan*

Having selected the Macintosh and microExplorer systems for our work, many additional decisions remained to select, configure, and integrate the routine computing environments for our users. The following summarizes this work.

#### <u>Text Processing - Editing</u>

There are many criteria for a system text editor including:

- Easy to learn
- Available from various contexts so that similar techniques can be used in editing mail, reports, and code.
- Powerful manipulation facilities allowing structures such as words, lines, paragraphs, pages, expressions, code blocks, etc. to be selected, moved, copied, reformatted, transposed, etc.
- Interchange ability allowing at least plain text to be imported from other systems and exported back to them.
- Extensibility in the form of keystroke macros and, ideally, customization libraries allowing us to write packages that make the editor "understand" a new kind of document structure.

Most of the commercially available Macintosh editors are targeted to desk-top publishing and so are fairly easy to use, but have manipulation facilities only for words, lines, paragraphs, sections, and pages. They are sadly lacking in understanding of other types of document structures such as programming languages or electronic messages. These editors typically offer interchange of "plain text" ASCII-only documents. They uniformly offer negligible extensibility.

Of the systems we've looked at, MicroSoft Word has proven most useful of those currently on the market, so we are using it in the meantime. Early demonstrations and tests of FullWrite Professional (marketed by Ashton-Tate) indicate that it may be superior to Word but its commercial release is just taking place. The non-commercial GnuEmacs might offer a complimentary solution as it is an outgrowth of the Emacs editor widely used in the AIM community. It offers familiarity and powerful extensibility, but it does not offer the easy-to-use interface and multi-font display expected on the Macintosh, and having two different editors would complicate matters.

In the coming year we plan to:

- Track new editor programs, seeking one that better meets our criteria
- Talk with editor vendors to encourage the addition of the desired features
- Further investigate GnuEmacs for the Macintosh II

### Text Processing - Aids

Most of the commercial text processing (TP) packages that we've looked at have built-in spelling checkers, sorting, hyphenation, forms filling, etc.

### Text Processing - Graphics

MacDraw, MacPaint, and the other Macintosh drawing programs offer state-of-the-art TP graphics capabilities. Pictures from these programs can almost always be integrated into a document being prepared with one of the commercial TP systems.

### Text Processing - Formatting

Most of the commercial TP systems are "What You See Is What You Get (WYSIWYG) editors, giving an on-screen representation of the final document during editing. The formatting quality and style control is quite good in the better systems, but we have found that some documents still require the extremely precise control offered by TeX and so are also using it, with it's "compile the manuscript with embedded commands" style interface.

### Text Processing - Bibliographic References

Another major shortcoming of the TP systems is the lack of automatic bibliographic reference generation and formatting. In writing scientific papers it is important to cite relevant work, and we have found it extremely useful to be able to extract the significant information from a large bibliographic database by placing a reference key where the citation should appear in the text. We are pursuing TP vendors in the hopes that they will implement this facility as well as investigating development of an auxiliary program to handle bibliography generation.

### Printing

Macintosh printing is fairly well developed in that most programs utilize the system-defined routines to print. We have installed PostScript on Imagen printers as well as Apple LaserWriters and are in the process of bringing up a spooling system on the file server.

Help Facilities

Most programs have built-in help on the Macintosh, as well as reasonably consistent interfaces, but this is not enough. We find that users are still confused so we are undertaking to produce short introductory documents to help users get started, and to point them in the right direction. We will investigate using HyperCard to organize this data.

System Information

As the system configuration has gelled we have begun thinking more and more about tools and protocols for getting information about the status of the overall system to the users and maintainers. We will need to address network loading, user location, resource usage (file space, printing, computing cycles), and status information for individual elements of the distributed computing environment. We also need access to personnel information, bulletin boards, and other shared databases.

Interpersonal Communication

See the section on electronic mail development

Systems Building Tools

We are developing expertise in the Macintosh Programmer's Workbench (MPW) environment, including with C and PASCAL, and HyperCard. We are also tracking Allegro Common Lisp, and Neuron Data's NExpert.

Filing

We plan to stay with the strategy of a few centrally located file servers, but the local disks on the Mac's complicate the system. The foremost concern is data backup. We have sketched out a design that would automatically copy new versions of documents (files) created on the Macintosh to a reliable file server (i.e., one that is backed up to tape). This backup program will allow for exclusion of some files (e.g., temporary files) and will make an effort to not have multiple copies of the same file on the server.

Also of significance to users who keep files on the Mac rather than a file server is the resultant inaccessibility from other computers. The proposed backup scheme would alleviate this problem as well. A full UNIX-based file backup and archival system is under consideration for the servers.

**3 - R&D Task Plan -- Update and Progress**

In the presentation to the site visit review team and Council, we layed out a detailed plan for our developments (see Figure 1). The following summarizes our pruning and reprioritization of those goals, based on the Council review, and progress this past year. In general, wherever we showed parallel developments to maintain capabilities among Xerox, Symbolics, TI, and other workstations, we have restricted our efforts to the Mac and Explorer environments, in accord with the Council recommendation for a focus of effort. While we continue to stay abreast of new workstation hardware and software, we have concentrated our system development work in the following areas for the Mac/Explorer environments. Progress in some areas has been limited by the reductions in systems manpower necessitated by NIH cuts in our award funding.

*3.1 - Remote Workstation Access*

- **TIP TCP-IP support**: We now have TCP-IP software running in our EtherTIPs.

- **Workstation TCP-IP access**: Each vendor has supplied the requisite software.

- **TELENET X.25/TCP-IP Ethernet Gateway**: The DEVELCON gateway has been installed and is used by the SUMEX-AIM community for TELENET access to the DEC 2060.

- **TELENET TELNET access (TCP-IP)**: The DEVELCON gateway is a bi-directional protocol translating gateway between X.25 and TCP-IP, and thus, fulfills this requirement.

*3.2 - Remote Virtual Graphics*

- **X Common Lisp client & server**: A Common Lisp X (CLX) client has been released for TI Explorers, SUN Workstations, and Symbolics. An alpha release of a CLX server is expected from TI this Summer. Since Xerox is moving its lisp environment to SUN workstations, and CLX runs on SUNs, we are not going to port CLX to Xerox D-machines. X runs under Apple UNIX (AUX) on MAC II's but is not implemented for the Apple Operating System (This latter operating system has its own graphics protocol, MacWorkstation, which we are experimenting with).

- **Common Window Application standard/Implementation**: We have not been able to give adequate attention to this item because of staff and budgetary constraints. It is worth noting that the Common Lisp User Environment (CLUE) is a window system defined on top of CLX and is currently in use in the KSL.

- **Develop/Extend Virtual Graphics applications:** Very little progress has been made in this area because of staff and budgetary constraints. We intend to emphasize the development of virtual graphics applications beginning this summer.

*3.3 - Distributed Mail System*

- **InterLisp mail reader/composer**: This software is completed, and is now part of the Lyric TCP-full-sysout. No further Xerox work is planned.

- **Redesign IMAP protocol (IMAP-2)**: This has been completed.

- **2060 IMAP-2 Server**: This has been completed.

- **UNIX IMAP-2 Server**: This has been completed, and is currently being alpha tested.

- **Common Lisp mail reader/composer**: The TI version should be completed and in initial testing by the end of June 1988. No work is planned on a Symbolics version. There are similarly no plans to translate the Xerox InterLisp client into Xerox Common Lisp given Xerox's plans to move to SUN's, and the existence of InterLisp within the Xerox environment for the foreseeable future.

- **Update Common Lisp IMAP-2 Clients**: This is completed.

- **UNIX mail client/reader/composer**: This project is on the backburner because of staff limitations, and the small number of SUN clients in use at SUMEX-AIM (a Macintosh-II client is underway for the Apple operating system rather than AUX, since the former is the primary OS in use at SUMEX-AIM). We have imported a UNIX version of the 2060 mail reader/composer called MM-C which was written at Columbia University. It is not a distributed mail system in that the reader/composer and mail file are assumed to reside on the same machine. The MM-C system will be used to provide national community and home mail services on the SUN-4 until further versions of the IMAP-2 clients are available.

- **Enhance reader/composer tools**: The reader/composer tools have undergone significant continual development since their release to our local user community. For example: message filtering was introduced earlier this year and one can now filter on free text searches; subject, from, to, cc and bcc text searches; new recent and old messages; On, before or since a given date; messages that are Seen/Unseen, Flagged/Unflagged, Answered/Unanswered, Deleted/Undeleted; and message keyword searches.

### *3.4 - General Computing Environment*

- **Lisp and AI shell environments:** Common Lisp now runs on Xerox systems using the Lyric sysout, and Lucid Common Lisp is on our SUN workstations. The Macintosh-II supports Coral Common Lisp. Finally, with the advent of the microExplorer co-processor on the Macintosh II, the entire Explorer Common Lisp environment is available on Macintosh II's configured with this board and an Ethernet interface.

  In addition, we have made considerable progress in analyzing the performance of Lisp systems on various kinds of hardware, with an eye toward guiding our work on future Lisp systems and the trade-off between specially microprogrammed Lisp machines and implementations on standard workstations. We have also defined the requirements for a powerful Lisp programming environment based on the key features of the Xerox, Symbolics, and TI environments that we use routinely in our AI research work.

- **Distributed File Support**: The Xerox Common Lisp RPC/NFS implementation has been completed, and Xerox has a strong interest in acquiring this software from us and including and supporting an enhanced version as part of their standard system release. We would in turn receive all improvements to the code. Both TI and Symbolics have released an RPC/NFS implementation. Because of budgetary and staff limitations this year, we have be unable to make any progress on *Network management, backup, and archiving tools.* This area has been given a high priority and we will begin to work on it this summer.

- **Distributed information access**: We have installed SUN UNIFY (a powerful relational data base system) on our SUN file server and implemented a Common Lisp remote procedure call/SQL query interface for Lisp machines to experiment with remote data base access. We have also been experimenting with the Apple HyperCard system for organizing and disseminating information in a distributed community.

- **Operations management tools**: We have made little progress in this area during this report period.

### *3.5 - Phasing of the transition to a distributed AIM community*

**Experiments in the Stanford/AIM community:** Each new piece of hardware or software has been tested initially by a selected subset of the Stanford/AIM community, e.g., members of the systems staff who are willing to put up with problematic software in the alpha test phase. It has then been beta-tested by a larger subset of the same community, and then released to any interested member of the community as a whole. A typical example is the Xerox IMAP Client, MM-D. After several months of alpha testing by two or three members of our systems staff, the software was distributed to other KSL research staff members for beta testing and suggestions for improvement. Finally, it became a part of the standard Xerox Lyric system used by the Stanford Knowledge Systems Laboratory.

## 4 - Remote Workstation Access, Virtual Graphics, and Windows

### *4.1 - Remote Access*

As we move towards a distributed workstation computing environment for AI research in the SUMEX-AIM community (and move away from the centralized, shared DEC 2060), a number of technical obstacles must be overcome. One of the most important is to eliminate the need for the user display to be situated close to the workstation computing engine. This is important in order to allow users to work on workstations over networks from any location -- at work, at home, or across the country. The first step has been getting reliable terminal access operational on all workstations. All workstations now have TCP/IP based terminal servers, and TCP/IP is being installed in the SUMEX network terminal concentrators. This allows primitive (non-graphical) access to the workstation's abilities. A more comprehensive access will be provided through our remote graphics work.

### *4.2 - Virtual Graphics*

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

As reported last year, we selected the MIT Project Athena X window system [4] as the remote graphics protocol standard for our work, and noted that X is a very complete protocol that has been developed over the past several years at MIT[1]. We also reported that an X *client*[2] for Texas Instruments Explorers was being written here at SUMEX-AIM, and that TI in conjunction with MIT was developing a *server*

[1] The X protocol was completely redefined last year. Its most recent version, X.11, is assumed in all of the discussion that follows.

[2] The *client* software runs on the Lisp machine and sends the graphics protocol commands to the remote user display system. The dual of the client is the X *server* software which runs on the user display system and translates the X protocol sent by a client Lisp machine into real graphics pictures and mouse actions.

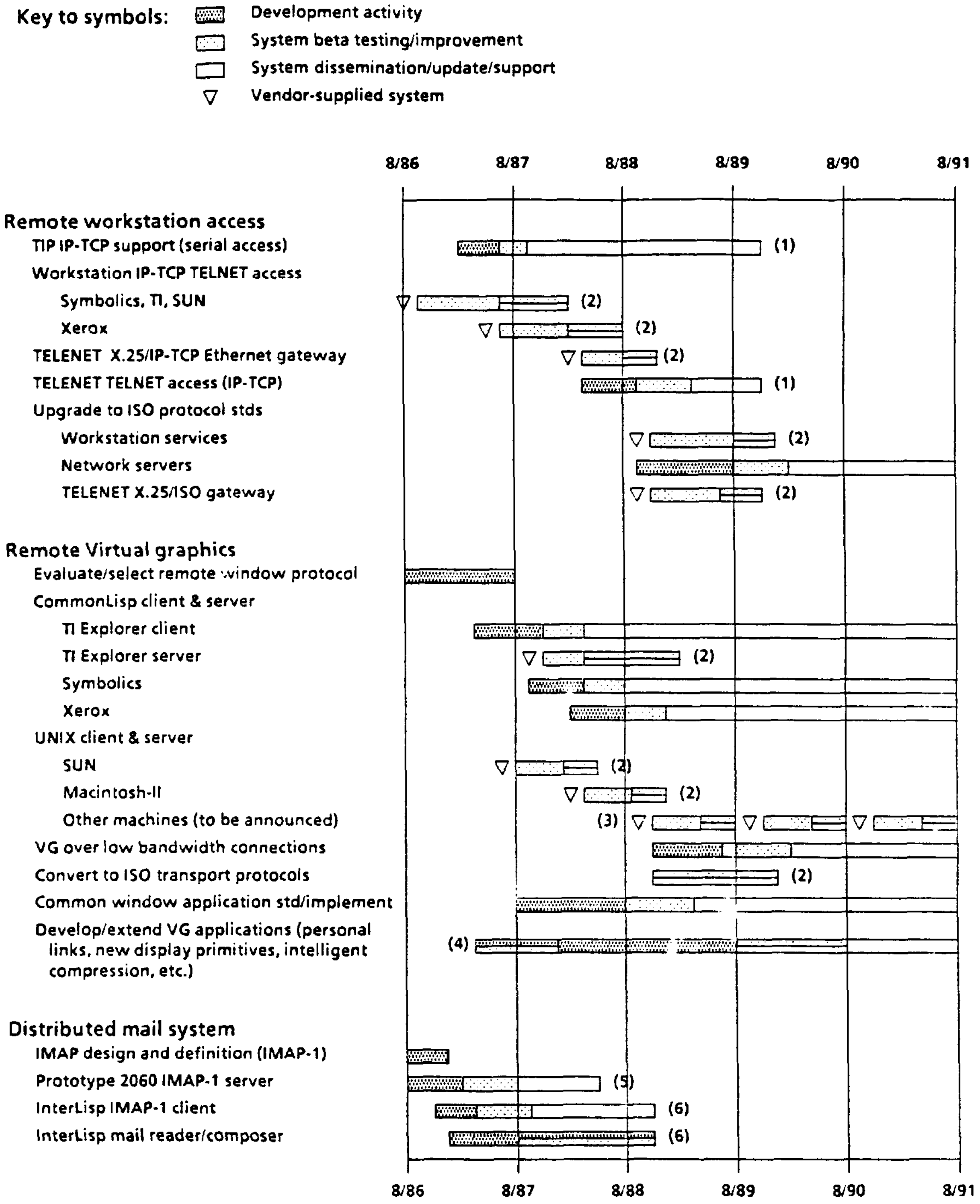


**Figure 1:** Core System Development Schedule

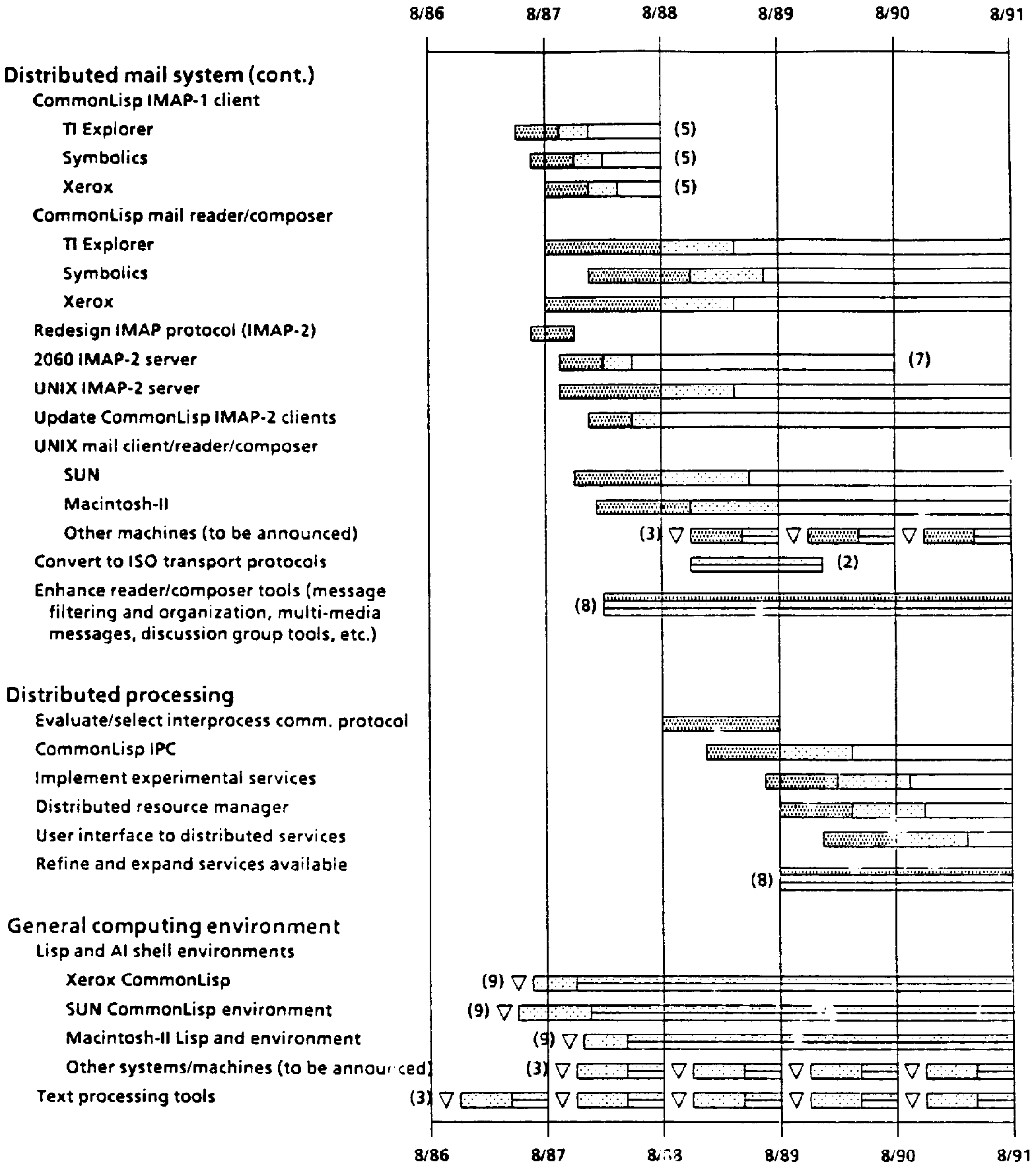
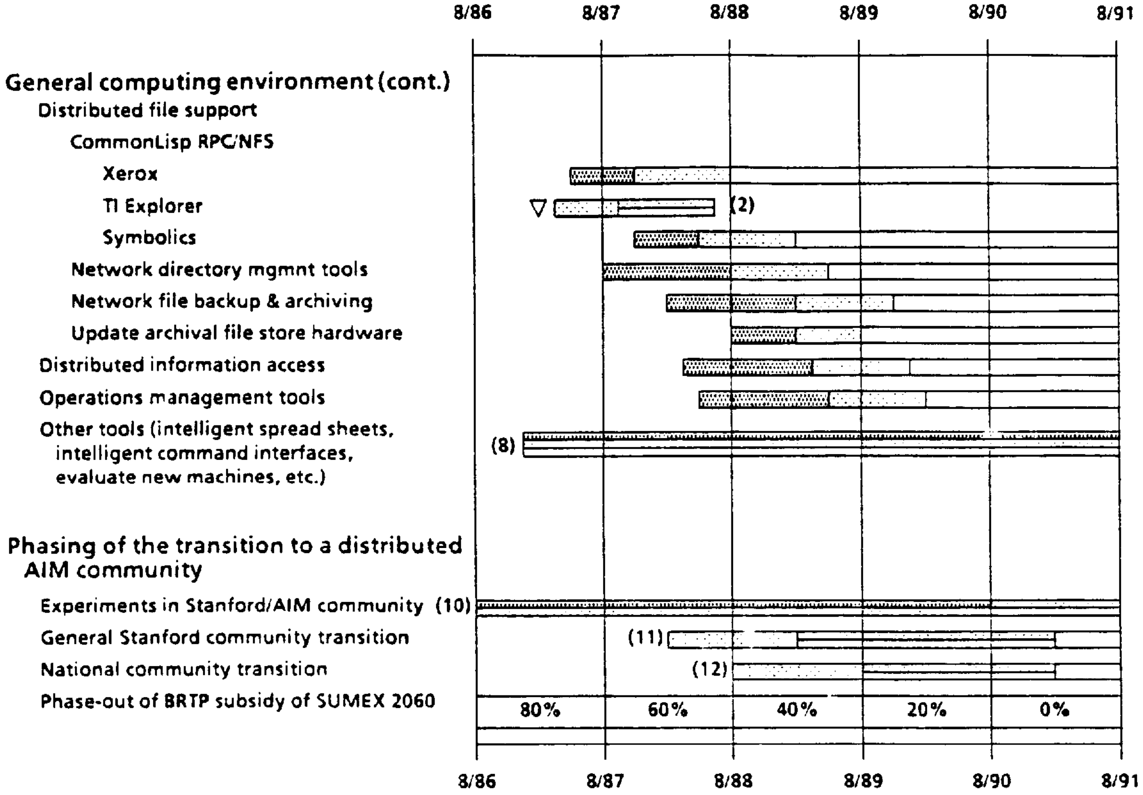


Figure 1: Core System Development Schedule, Continued



(1) We expect IP-TCP use to be dropped in favor of ISO protocols by late 1989.

(2) Vendor support expected after initial development/beta test period.

(3) Periodic review and evaluation of new vendors/workstations for use in the AIM community.

(4) Initial exploration of concepts followed by extensive development, testing, and dissemination.

(5) Convert IMAP-1 to IMAP-2 -- see below.

(6) Use of InterLisp is expected to phase out by late 1988 in favor of Xerox CommonLisp -- see below.

(7) We expect use of the DEC 2060 to phase out between the 4th and 5th years.

(8) This work will involve an extended period of development, testing, and dissemination, the exact schedule of which cannot be predetermined because it will depend on the results of other developments and experiments in this area.

(9) There will be an ongoing involvement with the vendor to test and support new system developments.

(10) We will use a small group of Stanford and AIM AI researchers and students to guide development and testing of distributed subsystems throughout the research period. Initially these will come mainly from the Stanford community which is easily accessible and already familiar with workstation use.

(11) We will need to have enough of the systems R&D done to begin testing tools in a broader Stanford user community.

(12) Dissemination to the general national AIM community will actually start earlier in an experimental mode (see 10) but full scale testing will begin in the 3rd year to prepare for a complete migration over the final 2 years.

Figure 1: Core System Development Schedule, Concluded

implementation. We had our X client well underway, when we discovered that TI and MIT decided to jointly develop a Common Lisp X-client (CLX) instead of a server.

Thus, in order to conserve our limited development resources, we chose to take a "wait and see" attitude, and redirect our programming efforts to the more immediate need of a distributed mail system. The time spent on our X-client was not wasted, since we gained a very deep insight into the protocol and its implementation, and planned to continue this project at a later date. In retrospect our choice was the correct one. CLX and a supporting CLX library for the creation of windows, menus, scroll-bars and other graphical objects is done, as is the implementation of a CLX-server which is due for alpha-release this June. Finally, The Common Lisp User Environment (CLUE) is now available from TI and is a window system on top of the CLX library which uses the Common Lisp Object System (CLOS). Thus, MIT and TI have provided a sufficient set of graphics primitives upon which one can build AI and systems tools in the X paradigm on TI Explorers. And, any other remote system which runs an X client and server will be able to have full duplex graphics communication with an Explorer. Currently, such a capability is available on Symbolics, SUNs, and VAXs.

Also, since CLUE is a more general window-system/user-environment on top of the CLX protocol primitives, it satisfies the long standing need for a portable window system for developing Common Lisp AI applications and will certainly find uses here at SUMEX-AIM.

Yet, much more work needs to be done in this area to fully develop and integrate remote graphics capabilities into Lisp machine systems in order to make low bandwidth connections a reality. We alluded to this topic in last year's report, and as yet, no one has begun to work in this area.

In the coming year we want to insure that cross-country connections will indeed support remote graphics and give usable response time. Success of this work will mean that one can use LISP machine systems from TELENET, ARPANET, or an EtherTIP connection throughout the SUMEX-AIM community.

### *4.3 - Remote Graphics Applications*

Our emphasis in the area of remote access/graphics has evolved from proving the concept of remote windows and remote tools to building real systems on top of remote access capabilities for routine use.

#### TALK

We've added a new protocol and new service to the TALK (intra-machine user communication tool which employs both text and graphics) system we described previously. We added the 'Sketch' service type as well as the ability to communicate using the IP/TCP protocols over the Ethernet.

Several groups in the national Xerox Lisp machine community who used the TALK program we distributed requested the TCP/IP capability. In addition to implementing the TCP/IP layer, since most BSD 4.* UNIX systems have a TALK facility, we wanted to make the TTY mode of our TALK compatible with the UNIX implementation. This would have allowed us to TALK to users on any VAX or SUN; we actually got to the point (using shortcuts for testing) where we could TALK from a VAX to a Xerox Lisp machine. Unfortunately, we ran into a problem in that the packets that SUN UNIX generates for TALK are different than those generated by a VAX due to differences in word alignment. This is true of UNIX running on other architectures as well as the

UNIX TALK program failed to make all of its data types network independent. We are, however, able to use all the TALK modes (TEdit, Sketch and TTY) to communicate between any Xerox Lisp machines available via the ARPA/Internet and have done a trans-continental TALK with a user at MIT.

We implemented the new 'Sketch' service for TALK which uses an object-based graphics editor as the basis for communication instead of a WYSIWYG text editor. The addition of this service was mostly a test to see if TALK was properly layered so that services could be added without modifying the TALK program itself, as we had designed. For the most part this was the case, the 'Sketch' service was brought up without modifying TALK in any way. Later we made some minor modifications based on what we learned to further simplify the addition of new services.

To put TALK into routine use in the KSL, we included it (and a few other Courier-based servers) in our default Lisp image so that it is always available to users in order to increase the use of sophisticated multi-machine tools.

## MacWorkstation

We have started a project based on the MacWorkstation software licensed by APPLE. The MacWorkstation program is designed to allow the Macintosh to start up programs on mainframes and receive graphic and text output to the MAC in a seamless fashion. We plan to write a Common Lisp window package (based on Common Windows) that uses MacWorkstation so we can connect (via Ethernet, Applenet or RS-232C/modem) to any of our Common Lisp engines and run the same piece of code on any and/or all of them. The MacWorkstation program uses a high level protocol which handles windows, menus, files, events, text and graphics and provides higher level access to the Macintosh graphic facilities than the native system library routines.

We do not plan to remote the existing window package of any given system; this interface will be written without regard to the window system on the machine on which it is being developed. We plan on providing a new library of window commands based on the emerging Common Windows standard. This will both simplify the design of the system and allow existing code that conforms to the CommonWindows standard to be used with little or no modification. This interface code can then be used on all Common Lisp engines (even mainframes) in our environment, including TI Explorers, SUN workstations, Symbolics and Xerox Lisp machines.

A Macintosh II is being used to run MacWorkstation and a Xerox Lisp machine is being used for the Common Lisp development. The two are currently tied together using a serial line which will eventually be upgraded to an TCP/IP Ethernet connection. We expect the resulting interface to work with the full range of Macintosh systems including the smaller ones typically in home use. It is possible to extend the capabilities of the MacWorkstation software on the Macintosh side, so once the basic system is in place we can then add whatever features we feel are necessary (if anything) to handle real Lisp applications.

We have already identified a group in our own laboratory that will be able to use the finished product. The SightPlan system needs to interface to BB1 output on an Explorer so that a user can see graphically how the site is being laid out. That graphics system is being implemented on a Macintosh since they can not use the Explorer screen which is full of BB1 data. We imagine systems based on the Common Windows/MacWorkstation interface that require the symbolic processing power of a dedicated lisp engine but also need to be available to a large number of users. We have test software running now which allows the Lisp machine to open windows on the Macintosh and do some simple graphics.

Other Remote Tools

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

### 5 - File Access and Management

A stable, efficient mechanism for storing and organizing data is central to any computing environment, and is one of the most challenging issues in the move to distributed, workstation-based computing. It is necessary to provide standard services, such as file backup, archival, a flexible, intuitive naming facility, and data interchange services (e.g., software distribution). Also, as the amount of data being manipulated grows, it will become more and more important to have powerful tools for managing hierarchies of files We plan to support the community with a number of UNIX-based file servers, like the VAX-based servers (see Figure 7) and the SUN-based server (see Figure 5) in use at SUMEX for several years. These will require continued SUMEX-AIM development, however. By keeping the number of servers small, the distributed namespace problem should be manageable in the near term. Current UNIX file servers are relatively cheap and fast. UNIX has many of the needed facilities, e.g., backup, long names, hierarchical directory structure, some file property attributes, data conversion, and limited archival tools.

However, while general issues of networking, remote memory paging services, and flexible file access have received considerable attention in both the academic and commercial development of file servers, there seems to be only slow development of other critical operational needs. For instance, the much-used file archiving system of the DEC 2060 (sometimes called off-line cataloged storage) has no analog service in the UNIX systems. Perhaps this is the result of UNIX having its origin in the small computer world where the number of users and volume of data has traditionally been quite low. Our efforts are going into trying to adapt a commercial system developed by UniTech to allow users to manage large file systems by providing access to and from off-line tape storage as well as maintain a historical archive of files. We have had a well structured organization for managing disk usage and accounting on our 2060 system and we plan to duplicate some of these features on the SUN-4 such that we can manage its usage in much the same way. A UNIX accounting system will be put in place to allow cost accounting and provide a quota enforcement capability on the distributed file server(s).

UNIX has always been weak in the management of large-scale file backup and with the advent of disks of near gigabyte size, it is clear that an organized approach to their usage is essential. In support of the long-term goals of the distributed community, we have been reviewing more advanced data storage methods. Optical disk systems are attractive but have not progressed as quickly as many had predicted. However, it seems likely that this sort of equipment would be ideally suited to satisfying our requirements for the archiving of files. Helical-scan magnetic tape equipment might also serve this function well. However, in both cases the absence of industry standards introduces some level of risk when planning to use these types of data storage equipment. We plan to continue our investigations of these technologies this coming year and to make a decision.

The AppleTalk/UNIX File Service package (AUFS) developed at Columbia University provides file support on a UNIX server so that Macintosh workstations can connect and see representations of the files present in the same icon-based format as the file system on the native Macintosh. File transfers are invoked by moving icons around the Macintosh screen just as if the UNIX server were an extension to the internal Mac disk. AUFS functions will be expanded to provide better and more service to our large Macintosh community.

### 6 - Electronic Mail

Electronic mail continues as a primary means of communication for the widely spread SUMEX-AIM community. As reported last year, the advent of workstations has forced a significant rethinking of the mechanisms employed to manage such mail in order to ensure reliable access, to make user addressing understandable and manageable, and to facilitate keeping the mail software distributed to workstations as simple, stable, and maintainable as possible. We are following a strategy of having a shared *mail server* machine which handles *mail transactions* with *mail clients* running on individual user workstations. The mail server can be used from clients at arbitrary locations, allowing users to read mail across campus, town, or country.

The mail server acts as an interface among *users*, *data storage*, and *other mailers*. Users employ a *mail access protocol* (MAP) to retrieve messages, access and change properties of messages, manage mailboxes, and send mail. This protocol should be simple enough to implement on relatively inexpensive machines so that mail can be easily read remotely. This is distinct from some previous approaches since the mail access protocol is used for *all* message manipulations, insulating the user client from all knowledge of how the mail is actually stored on the server. This means the the mail server can utilize whatever data storage and access methods are most efficient to organize the mail on a particular server system.

The first prototype Interim Mail Access Protocol (IMAP) was designed with this in mind. As noted in our previous report, we developed a prototype IMAP server on the DEC-20 and client on Xerox D-machines. The resulting mail environment proved to be quite usable and some D-machine users were able to use it as an alternative to the DEC-20 mail environment in their daily mail work.

During implementation of the prototype client we observed that the protocol had several deficiencies which made certain mail concepts difficult or impossible to implement. For example, there was no way to notify the client of newly arrived mail and inadequate extensibility in the protocol made it difficult to add such a functionality. Furthermore, it was a "lock-step" protocol with no mechanism for multiplexed operation, leading to synchronization problems. Finally, we rethought parts of our design based on our examination of several related projects, including MIT's PCMAIL and various POP2-based systems.

Our second-generation Interactive Mail Access Protocol (IMAP2) addresses these concerns. Instead of the lock-step query/response model of IMAP, IMAP2 uses tagged commands and data and explicitly allows unsolicited data to be sent from the server to the client. IMAP2 introduced a more formal structure to server-to-client path; in particular, all data is now identified unambiguously. IMAP2 also addressed various performance issues, such as allowing the fetching of multiple types of data items simultaneously. Furthermore, IMAP2 introduced server-based message searching and the concept of a structured "envelope" representing the information stored in the header of an RFC822 message.

An IMAP2 server manipulates the actual file store copy of the user's incoming electronic mail under direction from the IMAP2 client. As noted above, the client has no knowledge of the (possibly operating system dependent) format of mail on the server's file store; the IMAP2 protocol provides its own representation of mail and the server translates between this and its host system file store conventions.

The IMAP2 client issues a series of *fetch* commands to retrieve data from the server. A fetch command has two arguments: a *message sequence* and the name of the data item to be fetched. A message sequence can be a single message index, a range of message indices, or a list of numbers or ranges. A message index is one of a set of consecutive numbers which provide a quick pointer to that particular message in the mailbox. A mailbox with 20 messages uses indices 1 through 20.

As an example, a typical FETCH command might be

```
A0001 FETCH 2:7,10 ENVELOPE
```

meaning "fetch the envelope data for messages 2 through 7 and message 10". "A0001" is a random tag generated by the client, and used by the server in the completion response to an IMAP2 command. An envelope is in a format very much like a Lisp S-expression, and represents, in a structured fashion, data such as the From/To/cc/bcc lists, message Subject, Message-ID, etc.

There is presently no structured representation for the text of the message itself; the client must fetch the "RFC822" data attribute which is returned as a text string of the message in traditional RFC822 format. This is due to our immediate need for a system operationally usable in today's environment and also our feeling that the requirements for such a structured representation are not well-enough understood at this time.

There are other message attributes which can be fetched, the most interesting of which is the "FLAGS" attribute which contains various status flags associated with the message. The operation of removing a message from the mailbox consists of setting the system "\DELETED" flag for a message via the "STORE" command, and subsequently doing an "EXPUNGE" command.

Other operations in IMAP2 include LOGIN/LOGOUT for user authentication, SELECT for mailbox access, STORE to update/alter attributes associated with a message, EXPUNGE to permanently remove deleted messages from the mailbox, and SEARCH to do remote searches for messages based on various attributes.

The SEARCH command is quite sophisticated; for example the command

```
A0021 SEARCH FROM "SMITH" SUBJECT "REPORT"
      SINCE "20-MAR-88"
```

requests the server to return a list of message indices of all messages from SMITH with a Subject that includes the string "REPORT" that were placed in the mailbox after March 20, 1988. An elaborate set of remote search criteria have already been defined and implemented in the IMAP2 servers.

The server on the DEC-20 and Xerox Lisp client were upgraded to IMAP2. We have also implemented an IMAP2 server for UNIX, and have demonstrated a prototype client in Common Lisp for the Texas Instruments Explorer. The main thrust of our effort in the coming year, however, will be a client for the Macintosh.

### *6.1 - Xerox Lisp client*

An IMAP2-based client program for the Xerox Lisp environment was useable by mid-year. However, it was not in wide-spread use and lacked the necessary final polishing and debugging needed for release to a large community of users. A major push was made to make the necessary changes and the program was shuttled back and forth between members of the programming staff, who are specialists in different areas, in order to make the program as well-rounded as possible. Once the improved versions started to become available, the user community was steadily increased and where possible the resulting feedback was immediately reflected the next release. Along with the polishing of the client program, the documentation was completely revised and extended and the servers programs were further debugged.

Some of the specific changes to the Xerox Lisp client included the fixing of storage leaks, code optimization, basic window management and the splitting of the various command menus into several smaller menus and submenus, based on functionality. The major changes included the addition of the text compression and breaking features of the 'WEDIT' program (described in earlier reports) to the composition window to allow the mailer to freely convert between unbroken, streams of text and fixed length lines. Another major change was the addition of 'Zoom' mode for pulling together selected messages in the mail header browsing window. This allowed users to take full advantage of both the existing and extended text search capabilities that were added to the program without paying a penalty for retrieving excess mail headers from the server or doing manual searches via scrolling. A feature to search for mail files and allow the names of those files to be used as input to other commands with the click of a button was also added.

The Xerox Lisp client was further integrated into the KSL environment by adding it to one of the standard Lisp images (TCPFULL.SYSOUT), making it readily available to a larger community of users. The Xerox Lisp client is not going to be the mail client that the entire laboratory uses in the long term. The final push to finish the program was done in order to make available to a larger portion of the KSL a mail system based on the client and server model that everyone will be using eventually on other hardware. We wanted experience in running such a system and to allow users to have early feedback in the design of the ultimate system. The Xerox Lisp client will be maintained only as is reasonable as the mail system evolves so it can be used as a fast prototyping system for new ideas. The InterLisp-based IMAP2 layer of this client may be replaced by the Common Lisp one under development in order to reduce the maintenance costs of the various systems.

The Xerox Lisp client is currently the only fully functional, active client in our distributed mail system but probably will remain so for only a short while longer. Soon, we plan to make this client available to the Xerox Lisp user community along with the DEC-20 and UNIX IMAP2 servers.

### *6.2 - Texas Instruments Explorer client*

We are now writing an IMAP2 client for the TI Explorer system using Common Lisp. Although this project is not yet completed, it already can read mail, set message flags, move/copy messages between mailboxes and much more. Still to do is the message composing and sending. Except for the window system which is Explorer dependent, we expect much of this code can be exported to other Common Lisp workstations allowing us to develop mail systems for these workstations too.

### *6.3 - DEC-20 IMAP2 server*

The DEC-20 IMAP2 server was the first implementation of IMAP2 (and its predecessor, IMAP), and many of the ideas of our distributed mail environment were explored there first. This was due at least in part to the familiarity of the programmer with that traditional environment. The last major functionality addition to the DEC-20 server was the addition of the SEARCH facility last year. The current version of the DEC-20 server is a complete and robust implementation of IMAP2 and we do not anticipate any further work on it.

### *6.4 - UNIX IMAP2 server*

In our distributed mail system we expect to rely heavily on UNIX engines as servers. To this end we have written a UNIX IMAP2 Server (UIMS) for remote mail access. UIMS handles the full complement of IMAP2 commands and has been in alpha-test mode for the past month.

On a UNIX server a mailbox is represented by two files. The first is the actual text file containing the mail, and the second is an index file that caches a description of the text file's format. The index file is used to minimize UIMS disk i/o by keeping enough information about the text file and each message in that file to, first, speed random access to the text file, and second, make file access unnecessary for many of the IMAP2 protocol commands. For example, a binary representation of a message's date, flags and keywords is cached for each message. Thus queries on the state of these fields require no disk file access since a copy of the index file is kept in memory while a mailbox is selected. The index file also allows UIMS to implement a demand-paging algorithm to manage the core memory used for storing the text of each message as it is read or searched. As a consequence, only messages in which a user shows an interest need be kept in memory. In most instances this is only a small percentage of the number of messages in the text file. Such an algorithm conserves the server's resources, and is especially important under the UNIX operating system since it does not implement shared memory pages between processes.

For compatibility with MM-C, the Columbia University UNIX MM program (see below), UIMS and MM-C use the same text file for mail. MM-C has no knowledge of the index file associated with this file, and thus, UIMS will recreate the index file if the mail text file has been modified since UIMS last read it. Also, a locking mechanism is used to prevent simultaneous writes to the same mail text file. Unfortunately, locking under UNIX is not implemented in the operating system itself, and thus can be violated if a program not respecting file system locks attempts to write to a file that is open and locked by a program that does. All of the programs we have written and intend to write will respect these locks, as does MM-C.

UIMS also parses the MM-C init file, that describes a user's mail reading configuration, when this file exists. UIMS and MM-C have proved to be very successful when used in tandem for remote, and local mail reading, respectively.

### *6.5 - Transition Strategy and Plan*

While in transition from our mainframe-oriented paradigm to a distributed computing environment, the ability to read mail locally, from home or from geographically distant locations is clearly necessary at all points in this process. To make this move as easy as possible we hoped to be able to find a mail program that ran under UNIX to be used as a replacement for MM-20, the mail program now used at SUMEX-AIM by the majority of our community members. We were fortunate to discover MM-C, an MM-20 clone written at Columbia University in C to run on UNIX-based systems. We became an alpha test site for MM-C, and brought it up on both a VAX 11/750 and a SUN-3/180. It has been used extensively by members of our staff, and is purported to be an excellent mail program. By June of this year we should have installed MM-C on our new SUN-4 server.

As local community members move from the DEC-20 to the SUN-4, they will initially read their mail in the same mode as they did on the DEC-20, using MM-C if they do not have an Explorer or Xerox D-Machine. In the latter instances, they will use their respective clients in conjunction with the UNIX IMAP2 server discussed in the previous section. We anticipate that the SUN-4 will be used as a time-shared system for reading mail for the short term while the Mac client is being completed. When this is done, all such mail reading will be accomplished using the IMAP2 server/client paradigm.

In order to read mail from home one will dial-in to one of our EtherTIPs, TELNET to the appropriate server and run MM-C. The UNIX IMAP2 server is designed to be compatible with MM-C so that one can read their mail both locally and remotely without conflict.

It is also important to mention that Columbia University distributes MM-C with the statement "Permission is granted to any individual or institution to use, copy, or redistribute this software so long as it is not sold for profit, provided this copyright notice is retained". This makes MM-C an ideal mail reading program for the SUMEX-AIM community as we move away from the DEC-20, and towards our distributed environment, since SUMEX-AIM can freely distribute all of its mail-related software to its national community.

## 7 - System Building Tools

### *7.1 - Lisp System Performance*

One of the key issues in selecting the systems for our distributed computing environment was the performance of Common Lisp. In order to assist us in evaluating the performance of Lisp systems, we undertook an informal survey of Common Lisp environments using two KSL AI software packages, SOAR and BB1. The data collection for this evaluation is close to complete but we have only been able to do preliminary data analysis. The results are presented in Appendix B.

In this survey we have focused on execution speed for simulated system runs and for system compilation. However, we emphasize that execution speed benchmarks are only one aspect of the system performance evaluation, especially for Lisp systems. Other crucial issues like programming and usage environments, compatibility with other systems, ability to handle "large" problems, and cost must also be considered.

Both SOAR and BB1 were chosen because they are implemented in pure Common Lisp, making them extremely portable. SOAR is a heuristic search based general problem solving architecture developed by Paul Rosenbloom and BB1 is a blackboard problem solving architecture developed by Barbara Hayes-Roth. Neither of these systems is an intensive user of numeric computation. These systems were initially developed in environments other than those tested and no attempt was made to optimize their performance for any of these tests.

The workstation systems to be tested were chosen based on their availability as well as projected applicability in AIM community environments. Since we were interested in "real world" results, we ran the tests on each machine in what seemed to be its standard operating mode. The machines tested include:

- Apple Mac II
- DEC (uVAX II and III)
- HP 9000/350
- IBM (Compaq 386 "PS/2 model 80" and RT/APC)
- SUN (3/75, 3/60, 3/260, 4/260, 4/280)

- Symbolics 3645
- TI (Exp 1, Exp 2, uExp)
- Xerox 1186

and the Lisp systems running on these machines include:

- Explorer Lisp 3.0+ (4.0)
- Franz Extended Common Lisp 2.0
- HP Common Lisp 1.0
- Kyoto Common Lisp
- Lucid Common Lisp 2.x
- Symbolics Lisp 6.1
- VAXLisp
- Xerox Common Lisp (Lyric)

Analysis of the data that were collected is still underway but several conclusions are evident at a high level:

- The fastest machines for running BB1 (the microprogrammed TI Explorer 2 and microExplorer Lisp machines) are not the same machines that run SOAR fastest (SUN-4 RISC machines).

- Within a factor of two of the best performance, a considerable range of workstations based on stock microprocessor chips as well as specially microprogrammed Lisp chips have comparable performance.

Thus, while the "jury has been out" on the issue of special microprogramming versus stock instruction sets for Lisp systems, it now appears that stock systems have closed the performance gap. Unless substantial performance improvements are forthcoming from the specially microprogrammed architectures, it appears that Lisp-based research can just as well be done in "standard" workstation environments which would ease many issues of program export and integration.

### *7.2 - Lisp Programming Environments*

Even though performance gaps between microprogrammed Lisp systems and stock workstation implementations are narrowing, there still remains a significant difference in the quality of the development environments. The power of the development tools and environment is what has been the primary strength of Lisp machines, allowing rapid design, implementation, and debugging of complex programs. We believe the key to good development tools is integration, both in terms of consistency of interface, and in the ability to move seamlessly from tool to tool, carrying along appropriate data and state information. These qualities must be manifest in any KSL research computing system.

Over the years, KSL and AIM community AI systems have been implemented predominantly in the InterLisp, MACLisp, ZetaLisp, and more recently, Common Lisp dialects. Beginning in the early 1980's, our work moved from mainframe Lisp environments to workstation environments for many reasons, principally involving powerful tools for system development and debugging and graphical interfaces. Commercial versions of these tools, that evolved over many years in the Xerox D-Machine, Symbolics 36xx, LMI, and TI Explorer systems, have become an indispensable part of our work environment. Newer Lisp systems for workstations not specifically developed for Lisp have lacked many important features of these environments. Thus, in light of the runtime performance advances of stock workstations, we have attempted to summarize the key features of the Lisp machine environments that would be needed in stock machine implementations in order to make them attractive in a development setting.

The full version of this draft specification can be found in Appendix C. The following summarizes some of the key points about our effort to define the central environmental issues. We are continuing to seek comments from others in the AIM and broader AI communities as well as work with vendors and sponsors to try to get these feature integrated into commercially available Lisp systems.

- These requirements represent a snapshot of the tools and technology available on today's machines. AI has historically and will continue to ride the crest of the wave of new computing technologies for the foreseeable future, which enable ever more complex systems. Thus, these are not static requirements and we expect to be able to take advantage of the future improvements in hardware, graphics, and software as they are generated by computer science research and industry.

- We have tried to sort out the key features of current systems that are important to our research work. Except where explicitly stated, everything in the write-up (Appendix C) describes this "core" of functionality. Some items are clearly more important than others, but all represent needs that really guide our decisions about which new systems can be broadly used in the KSL.

- We have two overriding goals in adopting future computing environments (which may seem to be or actually are in some conflict). We want the most powerful development environments we can get to facilitate the building of complex AI programs. But at the same time, we want to be able to share (import and export) research results and tools with colleagues in other labs and so must maximize the portability of code among systems. We believe that these goals can be approached jointly by the establishment and careful adherence to standards where possible, while continuing systems development where necessary.

The requirements discussion is organized according to a "layered" view of Lisp environments, beginning with the upper levels. This organization is a conceptual framework within which to describe the various parts of the environment but may not correspond in full detail to the way system modules are actually organized. The elements of our description include:

- Program Development Tools and Environment
    - Editor
    - Debugger
    - Inspector
    - Software Management Tools
    - Performance Monitoring and Analysis
    - Lisp Listener

- Languages and Utilities
    - Windowing
    - Multiple Processes
    - Common Lisp
    - Compilation
    - Input/Output

  - Utilities
  - Interface Toolkit
  - Help System
  - Status Information
  - Printing
  - Pretty Printing

- Lower Level Issues

  - Address Space
  - Memory Management
  - Dedicated Versus Shared Systems
  - Hardware Capabilities
  - Overall System Integration in the KSL

### *7.3 - General System Building Environment*

Traditionally, a large set of languages and programming environments has been supported on the 2060 in order to encourage experimentation and development. We now believe that the experience gained in those years of broad experimentation can be distilled into a fairly small set of languages and tools, relieving the researcher of the need to learn many programming languages, while still providing the needed facilities to allow the experimentation to move further into the higher levels of knowledge representation systems and problem solving architectures. As we move to the workstation-based environment, we plan to phase out support for many of the languages we have offered in the past and concentrate on the most relevant languages for AI research and applications: C, Apple PASCAL, FORTRAN, InterLisp-D, ZetaLisp, and Common Lisp. Common Lisp has already achieved popularity as a standard (see page 74), and many projects are already using it. We expect to press for further adoption of Common Lisp as a community symbolic computing standard, consistent with prior investments in large software systems such as those which exist for on-going AIM projects. In addition, we will support important higher-level knowledge representation and problem solving architectures (e.g., S.1, KEE, Strobe, and others) as appropriate for community research and dissemination activities.

## 8 - Text Editing -- TMAX

One of the biggest obstacles in reducing our dependence on the 2060 is large document preparation, including sectioning, cross-referencing, and flexible bibliographic referencing. At the KSL this sort of processing has been done using SCRIBE (a registered trademark of Unilogic Ltd.), TeX, or LaTeX. There have been over fifty Xerox workstations (1108/9s and 1186s) in the KSL and as part of one experiment to move users off of the 2060, we worked to develop a system called TMAX (TEdit Macros And eXtensions) with the equivalent functionality of SCRIBE on these machines. This work is now finished and the following reports on the results of the experiment. The system has proven quite powerful for complex documents -- we know of two Masters theses that have been written using TMAX. We have turned the TMAX system over to Xerox where is now part of their standard working environment. The Xerox AIS group may use TMAX for their next set of InterLisp-D manual releases. Besides Xerox and Stanford, TMAX is in use at SRI International and Ford Aerospace.

The Xerox workstations come with a text editor called TEdit. TEdit is a WYSIWYG "text formatter" (which is part of what SCRIBE does) but it lacks the ability to do more complex operations. TMAX does not try to mimic SCRIBE; this would be a monumental task. Rather it extends TEdit by implementing some of the more commonly used features of SCRIBE. Currently there are five main areas of TMAX; numbering, table of contents, endnotes (similar to footnotes), forward and backward referencing, and indexing. There are three important points about the TMAX experiment. First it is completely menu-driven. To invoke any feature, the user simply moves the mouse to the appropriate item in the TMAX menu and buttons it. Secondly TMAX never changes the user's text. All the features are merely additions to the text. These additions may appear to be strings but they are really structured objects that are treated as single characters. This means users can delete any feature they add in the same way they would delete any other character. Finally, all the inserted features are shaded gray so users can distinguish them from the normal text.

We have written a twelve page help file on TMAX that uses all of the TMAX features. This document is both a description and demonstration of how to use TMAX. Below is a brief description of each of the TMAX main features.

### *8.1 - Numbering*

TMAX has an extremely flexible numbering facility that allows the user to number arbitrary objects such as chapters, sections, figures, examples, etc. The numbering scheme is completely up to the user. TMAX allows the user to create and edit a tree of "number nodes" where each node defines a level of numbering. The user also has the ability to define initial number values, the format of each number, text before and after the number (useful for headings), and the font of the number. To insert a number, the user simply buttons the appropriate node of the tree and TMAX figures out what the number should be and inserts it. Suppose for example the user is writing a book. The user could define the top node as Chapter with subnodes Section, Subsection, etc. Each time the user buttons Chapter in the number tree, the next monotonically increasing chapter number will be inserted in the document. It will also reset all Chapter's subnodes to their initial values. To insert a Section number, the user buttons Section in the tree and so on. The numbers can be ordinary Arabic numbers or upper and lowercase letters or Roman numerals. In the example above, the chapters could be "1.", the sections "1.A" and the subsections "1.A.i". Each value depends on the values of its immediately superior node in the tree. Finally, the numbering does not have to be inserted consecutively. If a user inserts a new number between other numbers, TMAX will automatically adjust all the numbers after the new one.

This numbering mechanism insures uniform numbering throughout the document. If at any time the user wishes to change the numbering format, he just edits the definition in the number tree and TMAX takes care of all the rest.

### *8.2 - Table of Contents*

Closely related to numbering is the "Table of Contents" facility. When this item is buttoned, TMAX creates a file containing all the numbers it has inserted along with any text the user specified before and/or after the number followed by a dotted leader and the page each number appears on. The file is sorted by page number and the user has the ability to exclude certain numbers from the file.

### *8.3 - Endnotes*

Endnotes are like footnotes in that they insert a superscript number but the text associated with the number is appended to the end of the document rather than the bottom of the page. When a user buttons the Endnote item, TMAX inserts the next monotonically increasing number and then prompts the user for the text string associated this the number. Like numbers endnotes need not be inserted consecutively. If an endnote is inserted between other endnotes, TMAX automatically adjusts all the endnotes after the new one. Finally, there is no correspondence between numbering and endnotes numbers.

### *8.4 - Forward and Backward Referencing*

Whenever a number or endnote is inserted, the user has the option of "tagging" it. At any point in the document, the user can reference this tag in one of two ways. The reference can be by the value of the number (e.g. "see chapter 5") or by the page the number is on (e.g. "see page 7"). Of course the user could reference the same number both ways and get something like "see chapter 5 on page 7". Whenever a tag is specified, it is added to a reference menu. If the user is referencing the same tag is several places, he can simply insert the reference from the menu rather than retyping the tag each time.

### *8.5 - Indexing*

TMAX supports two styles of indexing; simple and extended. With simple indexing, the user just specifies the phrase to be indexed. With extended indexing the user can 1) specify the phrase that will appear in the index, 2) specify the font of this phrase, 3) have the phrase sorted by some other phrase, and 4) optionally exclude the page number from the index (useful for index headings). Whenever an index is first specified, it is added to an index menu. If the user is indexing the same phrase is several places, he can simply insert the index from the menu rather than retyping the phrase each time. At any time the user can create an index file. This files contains the alphabetically sorted indices followed by the page numbers on which they appear. Optionally the user can request "manual style" indexing in which the page numbers are replaced by selected numbers. In this case the index specifies the chapter, section, etc. in which the index appears.

These are only the main TMAX features. There are several other minor features such as the ability to insert the date and/or time in several different formats.

## 9 - Distributed Information Resources and Access

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

The Macintosh HyperCard tool provides a very powerful environment in which to hierarchically organize and provide access to information in the form of text, graphics, pictures, and even sound. We have begun development of a "KSL stack" for HyperCard that will initially include descriptions of KSL and AIM personnel, SUMEX-AIM projects, SUMEX-AIM computing systems, KSL building maps, the KSL bibliography, etc. While the current version of HyperCard is a good environment for a prototype distributed information access system, several developments are needed from Apple for full utility. These include remote network access to stacks, allowing multiple simultaneous readers of stacks, and more flexibility of how stack "cards" are presented on workstation screens.

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

The initial implementation of the remote query system was done on a TI Explorer and then adapted to the Xerox D-machine Common Lisp environment. The query interface on the Lisp machine communicates with the Sun UNIFY database system via the Remote Procedure Call (RPC) mechanism which underlays the NFS remote file access system. The Explorer calls a server on the SUN and sends an SQL/DML query command as an argument to a remote query procedure, and receives the retrieved data and/or a message sent back from the server. SUN UNIFY can already manage multiple databases, so a client can have several databases open at the same time. The operations on the database are transaction-oriented, and therefore the concept of a database access session is applicable. The access functions currently implemented are *open a database*, *close a database*, *retrieve data from a database*, *insert records into a database*, *delete records from a database*, *update the database*, *lock a database*, and *unlock a database*.

### 10 - Distributed system operation and management

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

### 11 - Mainframe, Server, and Workstation System Environments

The various parts of the SUMEX-AIM computing environment require development and support of the operating systems that provide the interface between user software and the raw computing capacity. This includes the mainframe systems, network servers, and the workstation systems. Following are some highlights of recent system software environment developments.

### *11.1 - TOPS-20*

Our long-term plan to phase out the 2060 mainframe system has continued as scheduled. Development efforts on the 2060 have ceased, except where needed to keep the machine operational in the evolving distributed environment. This involves considerable work in areas such as file system archiving, retrieval, and backups; periodic updating, checkout, and installation of new versions of system software; the regular maintenance and updating of system host and network tables; and monitoring of and recovery from system failures, both hardware and software. Over the past year, the main areas of activity include:

- *TOPS-20 Domain Name Resolver Software* -- The Internet community has been in the process of converting to a domain naming scheme, to replace the flat address space of the old exhaustive host tables prepared by the Network Information Center. This past year, we worked very closely with a sister group at MIT tc 'ntegrate a version of the TOPS-20 Domain Name Resolver softwar with the TOPS-20 Mail System. In addition, several other network utilities, including TELNET and FTP were updated to use the Domain Resolver, rather than the old HOSTS.TXT table. Most other TOPS-20's are still running an ISI version of the DOMAIN resolver, which is inferior in several respects, so this represents a significant effort at SSRG to maintain network communication with the national Internet community.

- *Long-term access to SUMEX data files* -- The TOP-20 approach to providing references to off-line (archived) files presents substantial overhead to the active file system. Since SUMEX has always had an interest in preserving the history of the intellectual achievement resulting from our project, we are quite concerned with access to even very old data files. Thus we have prepared a staged scheme for retaining access to the files of former users. Though access is not as convenient to these files as it is to those archived by current users, there is, nonetheless, the ability to retrieve files of former users of our Tops-20 system and its predecessor, the PDP-10 based TENEX system.

- *Cost Center accounting* -- During the past year, the 2060 accounting programs were further updated to reflect the SUMEX Cost Center structure (see Page 121). Monthly reports have been promptly generated to reflect on-going usage. As part of the cost center management, a concerted effort has been made periodically to review all of the SUMEX accounts, and remove those that were no longer appropriate.

### *11.2 - UNIX*

We run the Stanford (SULTRIX) version of the UNIX system that is distributed by the University of California at Berkeley on our shared VAX 11/750 file servers. SULTRIX adds support for Sun Micro Systems Network File Service (NFS), and the PARC Universal Packet (PUP) protocol that is necessary for our D-machines. The two VAX systems are used extensively as file servers, and minimally as time-sharing systems. One of these systems (ARDVAX) is used for systems development and the UNIX version of the IMAP server has been written and debugged on it. Little system development effort is done on these beyond staying current with operating system releases and useful SULTRIX community developments.

### *11.3 - Sun File Server*

The SUN-3/180 file server (KNIFE), briefly mentioned in last year's report has been fully integrated into our distributed environment. In addition to providing file storage for about 50 users, it represents a prototype for the recently delivered SUN-4/280 in terms of implementation issues.

Imagen Host Software was installed to allow print spooling from the file server or any of its clients to any of the several Imagen Printers in the lab. This software was then updated to take advantage of the UltraScript (PostScript like document description language) on the Imagen 3320.

The InterLeaf Technical Publishing Software, and Frame Maker Software were installed for evaluation purposes.

The Unify Database Software, installed last year, continued to be used to administrative database applications.

The Columbia AppleTalk Package (CAP) was installed to allow local Macintoshes to access Apple-UNIX File Servers (AUFS) through Kinetics FastPath AppleTalk/Ethernet gateways.

A copy of the UNITECH UNIX file system backup/archival software was acquired for evaluation. The limited evaluation period ended before we could complete our evaluation, however, but we will probably try it again for our Sun-4/280 system, as an archival system is very important, and this package, although not perfect, looks like a good approach.

A 6250 BPI 1/2" magnetic tape Unit was added to the Sun-3/180 file server, and operational procedures put in place for regular file system backups.

### *11.4 - Xerox D-Machines*

A considerable, but shrinking part of the SUMEX-AIM community continues to use Xerox Lisp; primarily on Dandetigers and Doves (*Xerox 1109, 1186*). A donation of hardware from an industrial source increased the number of Dorados (*1132*) from 1 to 2.

The Xerox workstations proved reliable and economical to repair, averaging under $150 in repair costs per workstation for the year.

The *Lyric* release of the Xerox implementation of Common Lisp (which was beta-tested at SUMEX in March of '87 and saw general release in the Fall) proved so stable that it is still in use here a year later. The previous release, *Koto*, is still in use (by users of the commercial *KEE* and *Strobe* packages), but does not require or receive much support time from the staff. We will shortly beta-test the next release of Xerox Lisp--*Medley*.

We are also involved in the testing of a possible new hardware platform for the Xerox Lisp environment. The ONCOCIN system was used as a test program and the results of the experiments are encouraging. (A fringe benefit of the test is that research ONCOCIN has been upgraded from *Koto* to *Lyric* sooner than had been planned.)

The Lyric release required the reorganization of our font directories, containing several thousand files. (The font file names were changed to speed up lisp's search mechanism.) We deviated from the Xerox approach by further subdividing font directories according to font family name. The change required patching the lisp system software to exploit the new organization. This resulted in directories that are much easier to maintain and increased speed in font searches -- in some situations nearly twenty times faster.

As with any new lisp release we had to update the code and documentation for our over twenty *Lispusers* packages. This effort ranged from simple recompilation to serious rewriting to account for changes in the system and to add new features based on accumulated user feedback. New Lispusers packages we made available to the national community this past year included TCP and UDP time packages with both client and server routines, a new device driver for a color film recorder, client implementations of the Sun RPC and NFS protocols, the *MONITOR* package for viewing remote displays, and the *SKETCHTALK* package for remote graphic interaction between workstation users.[1]

We wrote an experimental Xerox Lisp *imagestream* driver for a Bell & Howell CDI-IV film recorder. This device was connected to a public 1108 via a 19.2 Kb serial line. In its HPGL (Hewlett Packard Graphics Language) emulator mode, the device was capable of drawing only single width straight lines in several colors. We used some bitmap-to-line drawing routines from our (previously implemented) plotter driver and a scheme using intermediate bitmaps (for the color planes) to produce a driver that could accurately render any image from the Xerox workstation on the film recorder (in color). This allows us to make photographic prints and slides of a fairly high quality. Patches to the Xerox *SKETCH* drawing program were necessary to make it work correctly in color mode. (These patches are applicable to other color devices.) The film recorder itself was donated by Bell & Howell and is on loan from the Computer Science Department.

We completed the Common Lisp implementation of Sun RPC (*Remote Procedure Call*) and NFS (*Network File System*) begun last Summer. The package was then used as part of a prototype system in which a database was accessible over the network to several types of workstation. Xerox has taken over responsibility for maintenance, improvement, and distribution of the package.

The Info-1100 discussion list which we sponsor continued to serve its readership on the ARPA Internet, Usenet, Bitnet, and CSNet. New subscriptions roughly equalled cancellations in number. Among the beneficiaries are other NIH-sponsored projects at Ohio State University and the University of Maryland.

The Xerox NS file server was reconfigured to run automatic nightly incremental backups from three of the T-300 disk drives to the file system on the fourth drive. Although this backup to disk does not give us the ability to retrieve files from the distant past (as tape backups do) it does give us the ability to restore a damaged file system to within a day of the damage. With three disk packs per drive and monthly full backups we are able to retrieve files from the previous 60 to 90 days. One problem with the previous system was automatic backup of Lisp sysouts (large binary images on the order of 3MB to 10MB in size), which tended to fill the incremental disk too quickly. To solve this, we reallocated disk space, moving sysouts to a directory on the backup drive itself. This kept the sysouts from being backed up incrementally. (They are still dumped monthly with the other drives and made available from a less active unit, reducing disk contention.) The removal of the sysouts from the user file system freed up large amounts of space and the user community on this server has increased in size and activity.

We moved the Xerox *Communications server* (which provides *Clearinghouse* service) to the MSOB machine room and upgraded its disk drive from 10MB to 40MB (using a

[1] The last two packages are described in the *Remote Workstation Access* section of this report.

disk from our spare parts supply). We did this for several reasons. By making it our NS boot server (instead of the Xerox printer) we were able to free up memory and disk storage on the printer so that we could turn on the the *reprinting* feature of the new server software. (This allows the printing of some Interpress documents that were too complex to print before the change.) We installed file service on the new boot server so that we could store the system installation files on it. Previously the boot and installation services were split between the printer and file server. Now we are able to boot the main file server over the network from software on the Communications server rather than using the slower, less efficient floppy-based system we were limited to previously. Finally, this gave us a Clearinghouse server on the MSOB network which makes authentication activities (like logins) much faster and more reliable (independent of network/gateway irregularities).

The Xerox University Grants Program (*UGP*) server maintenance agreement expires at the end of this reporting year (May 31st). This program covers the file server, 15 workstations and the print server provided in the original grant. The program has been funded to extend the warranties, but the corporation headquarters hasn't given the final go-ahead. If this does not happen, we will have to cover the cost of maintenance of the file server. In this contingency, we will maintain the workstations and printer ourselves (as with our other systems).

As reported last year, the Xerox NS file server software was upgraded to support random access of data. Although we had successfully written some patches to exploit this capability, we were fortunate to be asked by Xerox to beta-test an upgrade to Xerox Lisp's NS file system interface (to make use of the new features). This code is now part of our standard system and the file server is providing services such as hashed database access and on-line manual lookups (that were not possible before).

We also beta-tested the *ROOMS* multiple virtual display window environment (developed by Xerox PARC). This environment allows the user to have any number of virtual *screens* at his disposal. He can switch from one to another with ease and share windows among any subset he chooses. *ROOMS* may solve the display problems of having the ONCOCIN, OPAL and OPUS systems all in the same Lisp image. For easy access in the clinic, all three systems could be left set up (on their own screens) and, using a mouse click, the user could move between them.

### *11.5 - Texas Instruments Explorers*

The twenty Texas Instruments Explorers have enjoyed an increasing popularity as more projects have developed a need for the combination of execution speed, full Common Lisp, and sophisticated development facilities offered by the Explorer. Explorers have come into use in other parts of the national biomedical community as well, such as Ohio State University, MIT and the University of Maryland. However, the Explorer is still maturing as an AI workstation. Thus, our efforts have been directed at improving the environment of the Explorer by developing software, organizing user interest activities, and advising Texas Instruments.

Previous experience has shown that the greatest source of advancement for a particular computing environment is the user community. They are the most in touch with the deficiencies of the system, and thus uniquely positioned to address them, as well as to utilize the strengths of the system. The product developers of the system are frequently too involved in the lower levels of detail to produce general, effective solutions to problems, as well as being hampered by limited manpower resources. However, a significant amount of time and effort is required to organize this effort. This task has traditionally fallen to a user-run organization, such as DECUS or Usenix.

We have spearheaded an effort to organize an international users' group for the Explorer. The slightly misleading name for the group is NExUS, standing for National Explorer Users' Group. The goals of this undertaking are to:

- facilitate dissemination of information by organizing meetings where presentations and discussions can be used to make little-known techniques and facilities more widely known, as well as feeding back information on needs and wants to developers,
- allow more immediate communication via electronic mailing lists, which are used for distribution of important software fixes and discussion of items of general interest, such as new software tools, or proposed changes to the system,
- publish a periodic newsletter containing usage tips, salient extracts from the electronic mailing lists, and announcements,
- and, perhaps most importantly, establish and maintain a library of public domain, user supplied software.

A steering committee has been formed and set about forming a charter for the group, consolidating membership information, arranging a meeting in conjunction with AAAI '88, and settling the legal issues involved with setting up a code library.

There are already many entries ready for the library, most of which have been developed locally. We have maintained the software tools that were produced previously by fixing bugs, making improvements, and porting to new releases. Some of these have remained essentially the same, including:

- Analog Clock
- Backgrounds
- Backup To File System
- Batch Processor
- Choice Facility Enhancements
- Deexposed Mouse
- DEFSTRUCT Type Checking
- Development Tool Consistency Enhancements
- Finger Via TCP
- General Named Structure Message Handler
- Graph Viewer
- Graphical Value Monitors
- Imagen Via TCP
- Inspector Enhancements
- Map Over Files
- Rubber Band Rectangles
- Single Window VT100
- Snapshot Windows
- Soft Keys
- Source Code Controller
- Structure Enhancements
- Symbolics 36xx to Explorer compatibility package
- Transparent Windows
- Vertically Ordered Menu Columns
- Window Manager System Menu

Many of the tools have been enhanced or newly written this year.

General Inspector
This new tool integrates the three inspectors (data, Flavor, and Class) into one tool.

Search And Replace
This new tool allows looking for a pattern in a set of files, optionally replacing occurrences of the pattern with a new string.

Source Code Debugger
This new tool allows the user to compile code with "back pointers" so that the point of execution in the source can be shown in the debugger. This tool is a non-trivial achievement.

Spelling Checker
This tool is based on TI's spelling checker, but with extensions allowing user-defined dictionaries and other more minor enhancements.

Window Debugger Enhancements -- Additions to this tool include displaying SELF in the Locals pane even when it isn't explicitly referenced and allowing items in the Inspect pane to be modified.

Window Dragging
This new tool allows many windows to be "grabbed" by holding down the middle mouse button and dragged to new locations on the screen.

Window Icons
This new tools provides the infra-structure to allow shrinking temporarily unused windows to iconic representations.

Zmacs Enhancements
New features in this tool include:

- Commands to manipulate tag tables
- Commands to load tools and show their documentation
- Commands to talk to the compiler allowing one to see the result of optimizing code or to check the args in an expression
- A facility to spawn commands into a background process
- Commands to load and compile systems

Of course, all of these will be provided to the user's library, and many of them have already been given to other sites, including IntelliCorp, Berkeley, ISI, University of Maryland, and Ohio State.

Third party software is less utilized, but we stay abreast of the latest releases of the expert system shell KEE. We have installed a DVI previewing system from MIT allowing TeX and LaTeX output to be viewed on the Explorer screen. We have available several other imported tools such as a Common Lisp LOOP package and Macintosh style scrolling.

We have been following the development of the Common Lisp Object System CLOS closely. Several projects have already begun using Portable Common Loops (PCL), the public domain near-implementation of CLOS. We have done a number of things to make PCL easier to use on the Explorer, including making the debugger understand PCL forms and display them legibly, making the editor understand PCL forms and manipulate them properly, constructing a PCL class inspector similar to the Flavor inspector, and fixing bugs.

In addition to producing and maintaining these software tools, we attempt to provide extensive testing and evaluation of Explorer hardware and software products in a sophisticated university research environment in order that these products work more effectively when they are distributed to the national community. This testing is critical to the development of the computing environment since the combination of concentrated in-house expertise and close links to the product developers allows a turnaround on problem fixes unavailable in the broader scope.

This year we have participated in testing TI's high-end product, the Explorer II, Releases 3.1 and 3.2, Release 2.0 of the Network File System protocol implementation, and the new microExplorer system.

The microExplorer is an add-in co-processor board for the Apple Macintosh II based on TI's VLSI Lisp chip. After initial presentations from TI we were concerned that the microExplorer software would be too limited to allow many of the applications we envisioned for the system so we had a design review session with TI's developers before the final design of the system was frozen and as a result many aspects of the system are quite a bit more amenable to future extension. We were given one of the first systems outside of TI to beta test. During this testing dozens of problems were reported and addressed. It is our hope that this work will result in high-performance, low-cost sophisticated Lisp availability, allowing greater dissemination of AI software to the national community.

In addition to specific testing and evaluation, we are constantly finding, tracking, fixing, and reporting software bugs. This year we submitted twenty-nine new bug reports on Explorer system software, twenty of which had fixes included. All of these fixes have been made available to the national community in a patch file.

As well as working on these specific problems, we have had many meetings with Texas Instruments representatives wherein we have attempted to present the needs of the national community for short- and long-term AI workstation products, covering issues including the desirability of specialized hardware, address space, programming environment versus execution speed, and the ability to utilize the AI workstation's power for routine tasks.

Of course, there is also a large number of day-to-day activities needed to keep the computing environment pleasant, including resource management (e.g., disk space allocation, printer management), assistance with file backup and magnetic tape usage, and introducing new users to the system. We have produced documents targeted at complete novice users, users of InterLisp-D machines, and users of Symbolics machines in order to facilitate user education. These documents have been used as examples at various places in the national community.

For the coming year we plan to continue development and maintenance of the software tools, perhaps adding tools such as a DARPA Internet Domain Resolver, UNIX print spooling, better IP access control, CLX and CLUE packages, automatic backup, better documentation, better software management facilities, KSL specific NEW-USER facility, a Zmacs novice mode, and an Explorer version of the TALK program, as well as aiding the growth of the users' group.

### *11.6 - Symbolics*

*Symbolics*

Our work with Symbolics equipment has continued at a low level pending resolution of long-standing maintenance issues. As has been stated previously, in order for workstations to be competitive with time-shared mainframe computing resources, they must not only have a low purchase price, but must be cost-effective to maintain. This goal is normally achieved due to the economies of scale associated with having a large number of identical parts in an installation, as well as amortizing the cost of software development over many machines. We have come to reasonable agreements with all of the workstation vendors except for Symbolics. The high costs of service, the exceptionally high price of mail-in board repair, and the lack of a reasonable self-service alternative has left us unable to justify continued support of these machines unless a workable agreement can be reached. We have been trying a self-maintenance contract which is supposed to supply parts while we do the diagnosis and replacement work. This arrangement has not worked out well due to inadequate diagnosis software and procedures and delays in getting parts from Symbolics. In the coming year, we plan to try a standard Symbolics maintenance contract.

We plan to bring up Symbolics Genera 7.2 this summer. We have installed the REFINE system as well as KEE and FORTRAN.

### *11.7 - SUN Workstations and Lisp Environment*

Due to the high performance relative to purchase cost, Sun workstations drawn strong interest as Lisp engines. For the past year we have had three SUN 3/75 workstations in experimental use in the KSL. Because these were purchased for LISP work, we have added a 24 megabyte memory board (from Parity Systems Inc) to each of these. Also added were 70 megabyte SCSI disks. The systems have been set up to swap the large virtual memory to the local SCSI disk, rather than over the Ethernet to the server.

One of the systems is being used in the "Very Large Reusable Knowledge Base" project. A speech recognition box has been connected to another of the clients, and an interface to a Xerox InterLisp workstation running ONCOCIN developed to study the use of speech input for medical information. An evaluation of SUN workstations was made in terms of their suitability as a platform for a general physician's workstation which would support data management, analysis and display, and consultative software. For now, the SUN/UNIX environment was judged not to be competitive, in terms of cost or user interface technology, with other workstations environments for this purpose.

The vendor plans for LISP support on SUN workstations has be in a state of flux this past year. Currently their Lisp supplier is Lucid but there have been many rumors about a shift to Franz, Inc. as a replacement supplier. It appears that this uncertainty has delayed SUN's plan for improvements in their current package. However, since Franz Inc. is closely involved with CORAL in the LISP package we prefer to use with our MAC-II's, we might see a benefit in SUN making this change is suppliers.

Overall, Sun's Lisp still does not provide the programming environment power and maturity of the implementations for microprogrammed Lisp machines (see Appendix C). Sun seems to be devoting a good deal of effort to it's Symbolic Programming Environment (SPE) product to alleviate this situation. We examined an early version of SPE and found it very similar to it's foundation, the HyperClass system from Schlumberger. We are given to understand that SPE has made significant progress since those early versions, so we intend to re-evaluate it soon. We have been using the above mentioned HyperClass system as well as keeping new releases installed.

## 12 - Workstation Standards and Access

### *12.1 - Computing Environment Standards*

In a heterogeneous computing environment, such as AI research inevitably involves, the issue of cross-system compatibility is a central one. Users of various machines want to be able to share software, as well as be able to use various machines with a minimum of overhead in learning the operating procedures and programming languages of new systems. Thus, it is crucial to specify and propagate powerful, flexible standards for various aspects of the computing environment so that it is possible to transfer both skills and information among machines.

In order to improve the inter-machine compatibility of our software, we have been encouraging all users to use the Common Lisp programming language [5], as well as pressing vendors to provide more complete and efficient implementations of this language. We have already served as beta test sites for Xerox, Texas Instruments, and Lucid Common Lisp implementations.

The Common Lisp language, however, is only a subset of the software needed for our research. Research projects need higher-level powerful facilities, such as an object-oriented programming system and sophisticated error handling. Therefore we have been supporting and following the development of the Common Lisp Object System (CLOS) via membership in the electronic discussion group, technical contributions, and porting of Portable Common Loops (PCL), a predecessor of CLOS, to the TI Explorer. We are now encouraging vendors to produce efficient implementations of the system, and users to familiarize themselves with it. We are also encouraging vendors to adopt the proposed Common Lisp error system.

Other features of the computing environment also need to be standardized to be useful on more than one machine at a time. Another of the most important of these is the keyboard and display interface, often referred to as the "window system". See the virtual graphics section (page 48) for further discussion of window systems.

There are also many other areas which could benefit greatly from standardization, including document page description languages, text and graphics representations, and more networking protocols. However, it is important that standards not be entered into hastily, as an insufficient standard can often be worse than no standard at all. We intend to continue working to develop standards for these and other computing needs as the understanding of the issues involved matures.

### *12.2 - Protocol Standards*

Another important area of standardization has been inter-machine communication, or networking. Underlying all network I/O must be a network protocol for packet transfer between cooperating hosts. At SUMEX we have had long term experience with several such protocols; PARC Universal Packet protocol (PUP), TCP/IP, ChaosNet, and XNS/SPP. These have been used to implement higher level services such as remote booting, file transfers and access, TELNET access, electronic mail and bulletin boards, remote procedure call interfaces, remote graphics interfaces, and numerous utility services for locating network hosts, addresses, and the like. In the past we have elected to write servers for each new protocol in order to accommodate both vendor hardware and systems software requirements. This was necessary because no one protocol has been supported on all such systems.

With others in the computer science research community, we have pressed vendors to supply implementations of the DARPA standard TCP/IP communications protocols. We are pleased that the IP protocol family is now supported on all hardware and operating system configurations currently at SUMEX. And we expect to have IP support on any new systems we purchase in the future. Similarly, IP is supported on all of our UNIX based file servers, and the SUNet gateways route all IP datagrams. There has been a great deal of effort at Stanford and SUMEX to enforce IP as a standard protocol for new software development. This was motivated by its broad acceptance and the growing number implementations throughout the networking and vendor communities. This does not imply that we will abandon the other protocols but rather, since we are seeking to have *uniformity across all vendors* with the proposed Stanford distributed environment, we are choosing to limit new implementations to the IP protocol family. We are also currently working to provide improved support for TCP/IP in our Terminal Interface Processors (TIP's), having already implemented TCP/IP routing service.

Such standardization has a price, however, in that observed network communications speeds are often higher between equipment "tuned" to individual vendor protocols. We continue our efforts to assess performance bottlenecks and to refine system implementations to achieve acceptable throughput.

## 13 - Network Services

A highly important aspect of the SUMEX system is effective communication within our growing distributed computing environment and with remote users. In addition to the economic arguments for terminal access, networking offers other advantages for shared computing. These include improved inter-user communications, more effective software sharing, uniform user access to multiple machines and special purpose resources, convenient file transfers, more effective backup, and co-processing between remote machines. Networks are crucial for maintaining the collaborative scientific and software contacts within the SUMEX-AIM community.

### *13.1 - National and Wide-Area Networks*

#### 13.1.1 - General National Networking Issues

A 2-day conference on national research computer network issues was held in Washington, DC during April 1988. It was sponsored by The National Association of State Universities and Land Grant Colleges, The EDUCOM Networking and Telecommunications Task Force, and The New York State Education and Research Network (NYSERNet) with support from Apple Computer, IBM, and NYNEX. The meeting was well organized and encompassed views from academia, federal and state government, National Academy of Science/National Research Council (NAS/NRC) and White House Office of Science and Technology Policy (OSTP) study groups, industry, and Congress. There were discussions about needs and goals, NSFNet status and plans (including the Michigan MERIT/IBM/MCI implementation of the NSFNet backbone and regional network experiences, especially NYSERNet), DARPA's viewpoint, high bandwidth network research issues, a congressional view, and many funding/legal/technical issues. The meeting had over 300 attendees representing most of the major players. No one was at the meeting from NIH. Also, NIH has apparently not been very active on the FRICC (Federal Research Internet Coordinating Committee) set up by the OSTP for interagency coordination. The FRICC includes DARPA, NSF, DoE, NASA, and HHS.

Outside of DoD activities, almost all of the justification for the national NSFNet effort has come from the supercomputer initiative. Even most of the regional networks have been set up to facilitate access to supercomputers -- e.g., NYSERNet and BARRNet. The FCCSET (Federal Coordinating Council for Science, Engineering, and

Technology) report from OSTP is titled "A Research and Development Strategy for High Performance Computing. The NSF management overview of NSFNet goals mentioned the "workplace without walls" but the main motivating thrusts were access to supercomputers, central data bases (e.g., GenBank), the MOSIS facility for telemanufacturing, etc. Other uses for general research communication, collaboration, routine resource sharing are mentioned but do not capture primary attention. This balance of emphasis was all reemphasized in discussions with congressional representatives. Current federal funding covers the supercomputer effort and the NSFNet backbone but additional funds (estimated variously at $25M - $100M per year) to support a general National Research Internet (NRI) have yet to be argued for.

### 13.1.2 - NSFNet Status

The NSFNet consists of a national backbone network coupled to a number of regional networks (currently 11) that link research institutions with each other and with the backbone. The backbone is currently operating with 56 Kbit/sec lines and expects to move to full T1 (1.5 Mbit/sec) links in July this summer. The experimental network is being managed by MERIT, Inc. (a nonprofit consortium of 8 Michigan universities set up long ago in conjunction with the development of the Michigan timesharing system for IBM mainframes). In fact, IBM and MCI are research partners in the NSFNet project. The backbone will link the 6 current NSF supercomputer sites and 7 regional academic computing networks. NSF has put up $14M over 5 years and the State of Michigan has put up $5M from a strategic fund. IBM and MCI are contributing "materials and services".

There are longer term goals of making the configuration of the network more dynamically controllable depending on traffic needs and upgrading lines to DS3 service (45 Mbit/sec). The regional networks are not funded in this grant and many are facing funding crunches later this calendar year. Many regions are interstate and have bigger political barriers to cooperation, as compared to intrastate regional networks like BARRNet or NYSERNet -- although even these are coming under severe funding pressure. Some of the telephone companies (AT&T and MCI) think these services should be paid from from end user fees while many of the university people feel the regional intercampus links should be like the "national highway system" and supported out of general funds. An interesting suggestion was made that an analog to COMSAT might be set up for the NRI, since it will involve a close (nonprofit?) cooperation between federal, state, industrial, and academic interests.

### 13.1.3 - ARPANET Link

We continue our advantageous connection to the Department of Defense's ARPANET, managed by the Defense Communications Agency (DCA). A recent map of the ARPANET topology is shown in Figure 2. This connection has been possible because of the long-standing basic research effort in AI within the Knowledge Systems Laboratory that is funded by DARPA. Until the advent of NSFNet, ARPANET is the primary link between SUMEX and other university and AIM machine resources, including the large AI computer science community supported by DARPA.

Major changes are underway in the allocation and management of ARPANET resources. DARPA is seeking to control the funding it allocates to network operations and does not see itself as having the mandate of organizing or running an NRI. Effective May 1 of this year, the whole southern ARPANET route (1/2 of the total line miles!) was eliminated, forcing 6 major Texas universities to seek an alternative link with the Texas regional network (SesquiNet) for national

# ARPANET Geographic Map, 31 January 1988

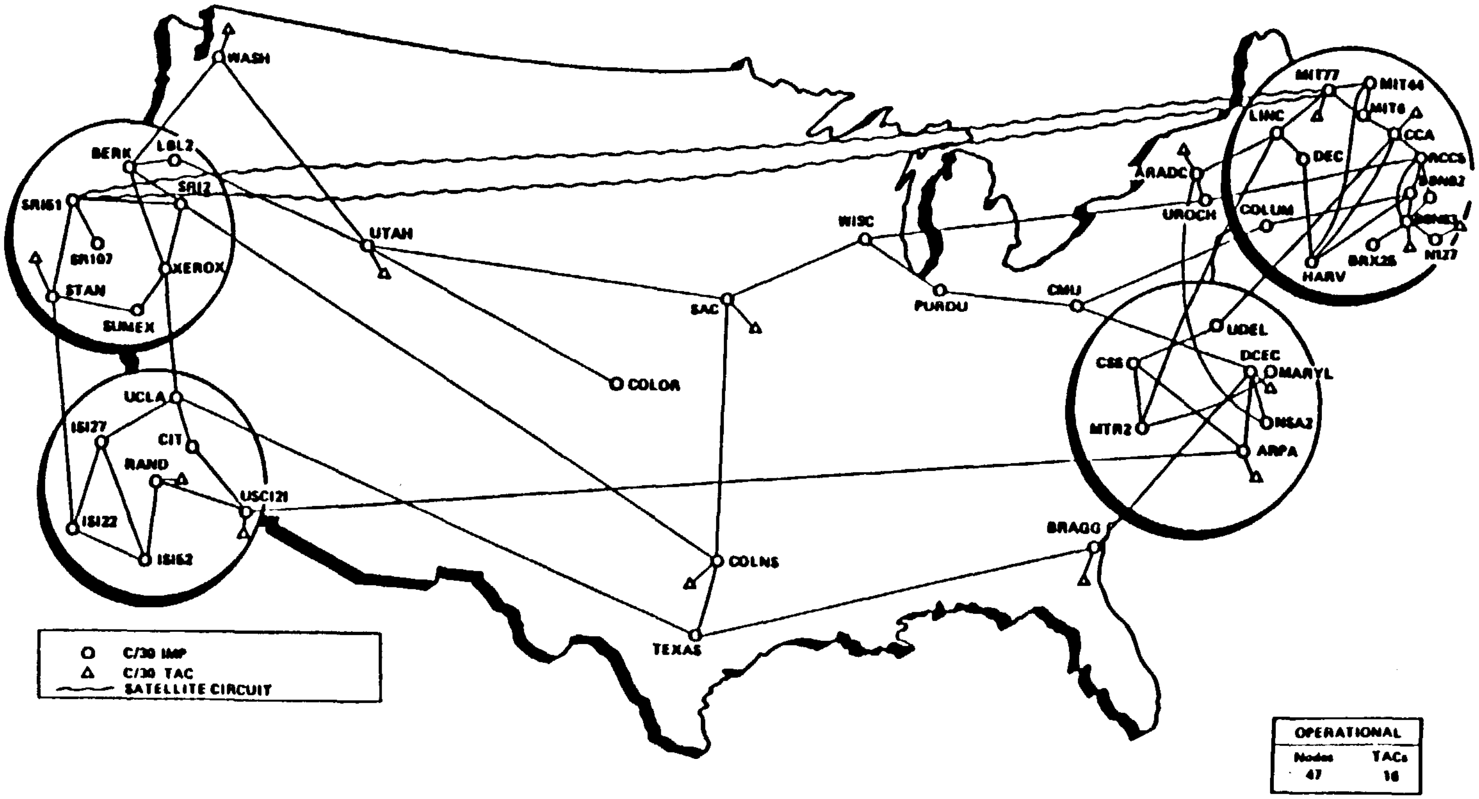


Figure 2: Recent ARPANET Coverage Map

communications through NSFNet. As a result of this cutback, the remaining 56 Kbit/sec ARPANET is becoming even more congested. By this summer, the ARPANET will move to a T1 backbone linking Boston, New York City, Washington DC, Pittsburgh, Chicago, Denver, San Francisco (NASA Ames), Los Angeles, and San Diego -- essentially 1 diagonal line across the US. Locations left off of this route will have to seek connectivity through NSFNet. The reduced coverage and higher speed but cheaper fiber optics lines will significantly reduce medium term network costs. "Multiple network connections" will be eliminated.

In the longer term, DARPA's role will apparently be limited to DoD communications research and support of its contractor community. Thus, the ARPANET will once again become very restrictive in terms of access and use. DARPA expects to construct a Defense Research Internet (DRI) over the next 3 years (in cooperation with DCA and SDI), with the current ARPANET disappearing after that. There is an explicit assumption in what DARPA is doing that the NSFNet or some other NRI will fill in the gaps. Thus, another "lead agency" will be needed. For now, NSF is serving in that role but there is no commitment by them or others that that will continue.

On the operational front, as a member of the ARPANET group, we have an obligation to help with certain network management tasks. For instance, we assisted ARPANET management with installing an additional 56 KB trunk line from our C/30 IMP to a new IMP at the nearby AMES Research Center (NASA). The addition provides an improved network topology in this area. We are working with the ARPANET group to reduce line costs and improve service in other respects also. We still depend heavily on the ARPANET to provide terminal communications service for many of our users as well as file transfer and electronic mail service.

### 13.1.4 - TELENET Link

The TELENET remains the primary connection mode for those national users without access to ARPANET services. As discussed below, we now have a more general interface to this network that provides an additional password check and outbound as well as inbound services. Because TELENET, like other commercial networks, is oriented to "local-echo" schemes, they do not put a high priority on echo-response across their network. Thus remote users are substantially inconvenienced when using programs requiring close interaction. It appears that our switch from TYMNET to TELENET (made about two years ago) has improved service slightly in this respect.

### 13.1.5 - X.25/Ethernet Link

The SUMEX-AIM transition to a distributed computing environment and our heavy use of Ethernet motivated our need for a connection between our Ethernet system and the Public Data Networks (specifically TELENET). Because our need involves Ethernet connections to other X.25 based hosts and Packet Assemblers - Disassemblers (PADs), sometimes called Terminal Interface Processors, as opposed to connections between two Ethernets via an X.25 link, we needed a device that provides protocol translation. Such a package would provide SUMEX with both an inbound and outbound capability relative to TELENET and users on SUMEX would be able to access the large variety of hosts and services on the PDNs (such as NLM and Dialog) in a simple and reliable manner. Though the high level protocols for file transfer and mail exchange are developing slowly in the X.25 environment, some progress is being made, so a general purpose interface to these networks is an important asset.

A substantial amount of effort went in to selecting and implementing this equipment. While a large number of vendors claimed to have TCP/IP-to-X.25 interfacing unit, these were generally encapsulating devices intended to connect two TCP/IP networks via an X.25 link. Instead, we needed a device which would provide protocol translation between the X.3 style terminal sessions of the X.25 networks and TCP/IP TELNET support. After a long and frustrating search among potential vendors (e.g., SUN, Bridge, and ACC), we located a suitable system manufactured by Develcon.

The experimental installation of the Develcon Network gateway has provided us with the capability to connect our Ethernet system to a variety of X.25 based networks (for terminal session support). This project has been pursued in conjunction with Stanford's Lane Medical Library and the Stanford University networking group (SUNet).

We have implemented a four-way gateway, connecting our Ethernet system with two X.25 point-to-point lines and the TELENET network (see Figure 8). The result has been a more flexible connection to the TELENET system (at the X.25 level) which gives users the ability to choose a variety of destinations on our Ethernet system when coming into our facility via TELENET. Unlike the simple X.25 PAD which this equipment replaces, the Develcon unit is able to provide both inbound and outbound services.

Our collaboration with Stanford's Lane Medical Library has resulted in this device providing access to the Tandem based Lane On-line Information System (LOIS) via Tandem's X.25 package (no Ethernet package is available), and a Lane sponsored X.25 leased line to the Bibliographic Retrieval Service (BRS) Inc's medical information system (Colleague).

Though this collaboration with the Lane Medical Library has been rather informal, it has proven valuable for both groups. As library science becomes heavily involved with computer based information services the interfacing of their systems with other computer based groups becomes very important. For instance, members of the SUMEX-AIM based Medical Information Sciences (MIS) program are able to make direct transfers of information from BRS to systems on our Ethernet.

The system supplier, Develcon Inc., has recently announced a product based on the work we did with them on this gateway. Further vendor development is needed to support improved accounting for system access and use and for user authentication and access control mechanisms.

### *13.2 - Microcomputer Networks*

We connected our Apple Macintosh computers in 2 buildings with *AppleTalk* and *PhoneNet* network products. More significantly, we integrated them with the rest of our equipment by connecting the microcomputer networks to the campus Ethernet networks using *Kinetics FastPath* gateways, a commercial spin off resulting from the SUMEX work on the SEAGATE gateway.

Software written at Columbia University, Stanford, and elsewhere, makes it possible for a Macintosh to share a VAX file server with the Lisp machines and to access hosts on the ARPA internet as a first-class workstation.

### *13.3 - Local Area Networks - LAN's*

In the past, we have developed local area networking systems to enhance the facilities available to researchers. Much of this work has centered on the effective integration of distributed computing resources in the form of mainframes, workstations, and servers. Network gateways and terminal interface processors (TIP's) were developed and extended to link our environment together. We are purchasing special purpose gateways to interface other equipment as needed too. This year we added several Kinetics Applenet-to-Ethernet gateways to support our Mac II's, LaserWriters, and Micro Explorers, and to link them with the LAN. The original Applenet-to-Ethernet gateways were developed at SUMEX-AIM. Kinetics licensed this software/hardware from Stanford University and used it as a basis for its product. A diagram of our local area network system is shown in Figure 9 and the following summarizes our LAN-related development work.

#### 13.3.1 - Ethernet Gateways

In our heterogeneous network environment, in order to provide workstation access to file servers, mail servers, and other computers within the Stanford local area network, it is necessary to able to route multiple networking protocols through the network gateways. As reported last year, the SUMEX gateways support PUP, Xerox NS, Symbolics/Texas-Instrument CHAOSNET, and the TCP/IP protocols. This support not only provides the routers necessary to move such packets among the subnetworks, but also other miscellaneous services such as time, name/address lookup, host statistics, boot strap support, address resolution, and routing table broadcast and query information. This year a device driver for the Micom-Interlan Ethernet interface was written. This interface has adequate on-board multibus memory to cache up to five 1514 byte input packets (the largest allowed on the network) and more than twenty small packets where "small" is less than 320 bytes. In contrast the older 3COM multibus interface has a fixed two buffer input cache. The larger packet cache minimizes dropped packets at the interface itself, and improves the performance of large file transfers that depend upon the ability to successfully send several back-to-back packets. In fact, we could not run NFS through our gateways with the 3COM interfaces unless we set the maximum number of back-to-back packets that the SUN-3 file server sends to two. With the Micom-Interlan interface, we could change this number to the default which is six, run NFS through the gateways, and achieve maximal performance.

#### 13.3.2 - Terminal Interface Processors

SUMEX-AIM has five TIPs, and in previous years we spent a reasonable amount of time maintaining and augmenting this software. Over the past year, this software has remained stable, and we plan on using vendor supported EtherTIP software for future TIPs. This software originated at SUMEX-AIM and has been commercialized by Cisco Systems, Incorporated, which licensed the EtherTIP/Gateway code from Stanford University.

Currently, one of our TIPs has ten dial-in ports and is used extensively by the local SUMEX-AIM community for dial-in access to the DEC 2060 and our VAX 11/750 servers from home during off hours. The four remaining TIPs are used to access various mainframes during work hours, and one of these runs the Cisco TCP/IP TIP code. As we begin to migrate from the 2060 to our distributed resource, we will in parallel replace our older PUP based TIP software with the above software. We currently run PUP in the majority of our TIPs because the 2060 more efficiently processes PUP, and most of our users login to the 2060 from these TIPs.

## 14 - Printing Services

Laser printers have become essential components of the work environment of the SUMEX-AIM community with applications ranging from scientific publications to hardcopy graphics output for ONCOCIN chemotherapy protocol patient charts. We have done much systems work to integrate laser printers into the SUMEX network environment so they would be routinely accessible from hosts and workstations alike. This expertise has been widely shared with other user groups in the AIM community and beyond.

SUMEX operates 7 medium-speed (8-20 pages per minute) Imagen laser printers, 2 low-speed (~3 ppm) Apple laser printers, and 1 low-speed (~3 ppm) Xerox laser printer.

Each of the Imagen printers includes an emulator for a line printer, a daisy wheel printer, a Tektronix plotter, and a typesetter (using the *Impress* language). Additionally, the two Imagen *3320* printers implement the *PostScript* typesetter language (also implemented by the Apple LaserWriters) required for printing Macintosh documents. The Xerox printer (an *8046*) interprets the *InterPress* typesetter language.

In total, the laser printers printed about half a million pages of output during the year. Most of the printout was simple text, followed in quantity by formatted text in *Impress* format, Impress-format drawings, and screen dumps. About 16,000 pages each of PostScript-format drawings and formatted text were printed on the Apple LaserWriters and Imagen 3320's (an eightfold increase over last year).

We were one of a small number of sites to beta test Imagen's implementation of the PostScript typesetter language. We submitted a number of bug reports and all the bugs we found were fixed in short order. (Imagen went on to become the first printer manufacturer to ship a non-Adobe implementation of PostScript.) In consideration of our help with the test, Imagen transferred ownership of the test hardware to our lab.

Since the proliferation of Macintosh computers demands higher-speed printing than the Apple LaserWriter can provide, another Imagen 12/300 was upgraded to model 3320 (configured for PostScript) and installed in the other building inhabited by SUMEX and KSL users. Imagen granted a special discount on this upgrade in consideration of our dissatisfaction with the longevity of the 12/300 printers purchased earlier.

## 15 - General User Software

We have continued to assemble (develop where necessary) and maintain a broad range of user support software. These include such tools as language systems, statistics packages, vendor-supplied programs, text editors, text search programs, file space management programs, graphics support, a batch program execution monitor, text formatting and justification assistance, magnetic tape conversion aids, and user information/help assistance programs.

A particularly important area of user software for our community effort is a set of tools for inter-user communications. We have built up a group of programs to facilitate many aspects of communications including interpersonal electronic mail, a "bulletin board" system for various special interest groups to bridge the gap between private mail and formal system documents, and tools for terminal connections and file transfers between SUMEX and various external hosts. Examples of work on these sorts of programs have already been mentioned in earlier sections, particularly as they relate to extensions for a distributed computing environment.

At SUMEX-AIM we are committed to importing rather than reinventing software where possible. As noted above, a number of the packages we have brought up are from outside groups. Many avenues exist for sharing between the system staff, various user projects, other facilities, and vendors. The availability of fast and convenient communication facilities coupling communities of computer facilities has made possible effective intergroup cooperation and decentralized maintenance of software packages. The many operating system and system software interest groups (e.g., TOPS-20, UNIX, D-Machines, network protocols, etc.) that have grown up by means of the ARPANET have been a good model for this kind of exchange. The other major advantage is that as a by-product of the constant communication about particular software, personal connections between staff members of the various sites develop. These connections serve to pass general information about software tools and to encourage the exchange of ideas among the sites and even vendors as appropriate to our research mission. We continue to import significant amounts of system software from other network sites, reciprocating with our own local developments. Interactions have included mutual backup support, experience with various hardware configurations, experience with new types of computers and operating systems, designs for local networks, operating system enhancements, utility or language software, and user project collaborations. We have assisted groups that have interacted with SUMEX user projects get access to software available in our community (for more details, see the section on Dissemination on page 123).

## III.A.2.5. Relevant Core Research Publications

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

HPP 80-29
H. Penny Nii; **An Introduction to Knowledge Engineering, Blackboard Model and AGE,** March 1980, **45 pages**

KSL 86-19
J.P. Rice; **Poligon, A System for Parallel Problem Solving,** April 1986. To appear in: *DARPA proceedings, Asilomar 1986.* **16 pages**

KSL 86-20
J.R. Delaney; **Multi-System Report Integration Using Blackboards,** March 1986. Submitted for publication to: *1986 American Control Conference.* **12 pages**

KSL 86-38
**STAN-CS-87-1147.** Barbara Hayes-Roth, M. Vaughan Johnson Jr., Alan Garvey, and Micheal Hewett; **A Modular and Layered Environment for Reasoning about Action,** April 1987. To appear in: *The Journal of Artificial Intelligence in Engineering, Special Issue on Blackboard Systems, October 1986.* **63 pages**

KSL 86-41
H. Penny Nii; **CAGE and POLIGON: Two Frameworks for Blackboard-based Concurrent Problem Solving,** April 1986. To appear in: *DARPA Proceedings, Asilomar 1986* **8 pages** [Superceded by KSL 87-71]

KSL 86-62
**STAN-CS-87-1175.** David C. Wilkins, William J. Clancey, and Bruce G. Buchanan; **Using and Evaluating Differential Modeling in Intelligent Tutoring and Apprentice Learning Systems,** January 1987. To appear in: *Intelligent Tutoring Systems: Lessons Learned, Lawrence Erlbaum Publishers, 1987.* **37 pages**

KSL 86-63
David C. Wilkins; **Knowledge Base Debugging Using Apprenticeship Learning Techniques,** October 1986. **15 pages**

KSL 86-69
**STAN-CS-86-1136.** Harold Brown, Eric Schoen, and Bruce Delagi; **An Experiment in Knowledge-Base Signal Understanding Using Parallel Architectures,** October 1986. To appear in: *Parallel Computation and Computers for AI, J.S. Kowalik Editor, Kluwer Publishers.* **39 pages**

KSL 87-27
**(Journal Memo)** Gregory F. Cooper; **Probabilistic Inference Using Belief Networks Is NP-Hard,** August 1987. **23 pages**

KSL 87-28
**(Journal Memo)** Eric J. Horvitz; **A Multiattribute Utility Approach to Inference Understandability and Explanation,** September 1987. **34 pages**

KSL 87-34
**(Working Paper)** Russell T. Nakano; **Experiments with a Knowledge-Based System on a Multiprocessor: Preliminary Airtrac-Lamina Qualitative Results,** June 1987. **72 Pages**

KSL 87-36
Lawrence J. Selig; **An Expert System Using Numerical Simulation and Optimization To Find Particle Beam Line Errors,** May 1987. **39 pages**

KSL 87-38
Naomi S. Rodolitz; **Guidon Manage - Tutoring for Strategic Knowledge,** June 1987. **43 pages**

KSL 87-39
**(Journal Memo)** Glenn D. Rennels, Edward H. Shortliffe, Frank E. Stockdale, Perry L. Miller; **Updating an expert knowledge base as medical knowledge evolves: Examples from oncology management,** June 1987. To appear in *Proceedings of the AAMSI Congress 87.* **6 pages**

KSL 87-40
Alan Garvey and Barbara Hayes-Roth; **Implementing Diverse Forms of Control Knowledge in Multiple Control Architectures,** June 1987. **32 pages**

KSL 87-41
Thierry Barsalou; **An Object-Based Interface to a Relational Database System,** June 1987. **10 pages**

KSL 87-43
**STAN-CS-87-1166.** Hiroshi G. Okuno and Anoop Gupta; **Parallel Execution of OPS5 in QLISP,** June 1987. Shorter version to appear in: *Proceedings of the Fourth Conference on Artificial Intelligence Applications (CAIA-88), IEEE, March 1988.* **18 pages**

KSL 87-44
**STAN-CS-87-1178.** Gregory T. Byrd, Russell Nakano, and Bruce A. Delagi; **A Dynamic, Cut-Through Communications Protocol with Multicast,** August 1987. **23 pages**

KSL 87-45
**(Journal Memo)** David Heckerman and Holly Jimison; **A Perspective on Confidence and Its Use in Focusing Attention during Knowledge Acquisition,** July 1987. To appear in: *Proceedings of AAAI 87.* **10 pages**

KSL 87-46
**(Journal Memo)** Michael P. Wellman and David E. Heckerman; **The Role of Calculi in Uncertain Reasoning,** July 1987. To appear in: *Proceedings of AAAI 87.* **12 pages**

KSL 87-48
**(Journal Memo)** Gregory F. Cooper; **Computer-Based Medical Diagnosis Using Belief Networks and Bounded Probabilities,** February 1988. To appear in *Topics in Medical Artificial Intelligence edited by Perry Miller.* **17 pages**

KSL 87-49
**(Journal Memo)** Mark E. Frisse and Gio Wiederhold; **Retrieving Information from Hypertext Systems,** August 1987. **14 pages**

KSL 87-50
**(Journal Memo)** Mark E. Frisse; **Searching for Information in a Hypertext Medical Handbook: The Washington University Dynamic Medical Handbook Project,** August 1987. **19 pages**

KSL 87-51
**(Journal Memo)** Gregory F. Cooper; **Expert Systems Based on Belief Networks - Current Research Directions,** February 1988. **21 pages**

KSL 87-52
Curtis P. Langlotz; **Advice Generation in an Axiomatically-Based Expert System,** August 1987. To appear in: *Proceedings of the Eleventh Annual Symposium on Computer Applications in Medical Care, Washington, DC, 1987.* **8 pages**

KSL 87-53
**(Journal Memo)** David Heckerman and Eric J. Horvitz; **On the Expressiveness of Rule-Based Systems for Reasoning with Uncertainty,** August 1987. To appear in: *Proceedings of AAAI, Vol. 1, July, 1987.* **7 pages**

KSL 87-54
Joshua Lederberg; **How DENDRAL Was Conceived and Born,** August 1987. To appear in: *Proceedings of ACM Symposium on the History of Medical Informatics, National Library of Medicince, November 1987.* **19 pages**

KSL 87-57
**STAN-CS-87-1184.** Hiroshi Okuno, Nobuyasu Osato and Ikuo Takeuchi; **Firmware Approach to Fast Lisp Interpreter,** September 1987. To appear in: *Proceedings of Twentieth Annual Workshop on Microprogramming (MICRO-20), ACM, December 1987.* **25 pages**

KSL 87-58
William J. Clancey; **Knowledge Engineering Methodology: An Annotated Bibliography of NEOMYCIN Research,** September 1987. To appear in: *R. Nossum (ed.), Lecture notes in Computer Science - ACAI'87, Springer-Verlag, 1988.* **10 pages**

KSL 87-59
Janet McLaughlin; **Utility-Directed Presentation of Simulation Results,** December 1987. **57 pages**

KSL 87-60
Richard M. Keller; **Defining Operationality for Explanation-Based Learning,** October 1987. To appear in: *Artificial Intelligence Journal.* **19 pages**

KSL 87-61
**STAN-CS-87-1188.** Russell Nakano and Masafumi Minami; **Experiments with a Knowledge-Based System on a Multiprocessor,** October 1987. **47 pages**

KSL 87-62
John A. Brugge and Bruce G. Buchanan; **Evolution of a Knowledge-Based System for Determining Structural Components of Proteins,** October 1987. **26 pages**

KSL 87-63
**(Journal Memo)** Leslie Elaine Perreault; **Automatic Test Case Generation by Modeling Patient States and Physician Actions,** October 1987. **30 pages**

KSL 87-64
**(Journal Memo)** Eric J. Horvitz; **Problem-Solving Design: Reasoning about Computational Value, Tradeoffs, and Resources,** November 1987. To appear in: *Proceedings of the National Aeronautics And Space Administration Artificial Intelligence Forum, Mountain View, CA.* **19 pages**

KSL 87-65
**STAN-CS-87-1189.** Bruce A. Delagi, Nakul Saraiya, Sayuri Nishimura, and Greg Byrd; **Instrumented Architectural Simulation,** November 1987. **7 pages**

KSL 87-67
Barbara Hayes-Roth; **Dynamic Control Planning in Adaptive Intelligent Systems,** November 1987. **7 pages**

KSL 87-73
Stephen Barnhouse; **Knowledge Base Tours: Introducing a Knowledge Based System to a Novice User,** December 1987. **15 pages**

KSL 88-01
M. Vaughan Johnson and Barbara Hayes-Roth; **Learning to Solve Problems by Analogy,** March 1988. **15 pages**

KSL 88-02
H. Penny Nii, Nelleke Aiello, and James Rice; **Frameworks for Concurrent Problem Solving: A Report on Cage and Poligon,** March 1988. **28 pages**

KSL 88-04
J. P. Rice; **Problems with Problem-Solving in Parallel: The Poligon System,** January 1988. **21 pages**

KSL 88-06
**(Thesis)** Mark Alan Musen; **Generation of Model-Based Knowledge-Acquisition Tools for Clinical-Trial Advice Systems,** January 1988. **293 pages** [ONLY ABSTRACT AVAILABLE]

KSL 88-09
**(Working Paper)** Bruce G. Buchanan and Reid G. Smith; **Fundamentals of Expert Systems,** March 1988. **33 pages**

KSL 88-10
Gregory T. Byrd and Bruce A. Delagi; **A Performance Comparison of Shared Variables vs. Message Passing,** February 1988. To appear in: *May 1988 ISI Supercomputing Conference Proceedings.* **15 pages**

KSL 88-11
Richard M. Keller; **Operationality and Generality in Explanation-Based Learning: Separate dimensions or opposite endpoints?,** February 1988. To appear in: *Proceedings of the AAAI Symposium on Explanation-Based Learning, March 1988, Stanford, CA.* **5 pages**

KSL 88-12
Clifford E. Wulfman, Ellen A. Isaacs, Bonnie Lynn Webber, and Lawrence M. Fagan; **Integration Discontinuity: Interfacing Users and Systems,** February 1988. **12 pages**

KSL 88-13
Eric J. Horvitz, John S. Breese, and Max Henrion; **Decision Theory in Expert Systems and Artificial Intelligence,** February 1988. To appear in: *Journal of Approximate Reasoning, Special Issue on Uncertainty in Artificial Intelligence, July 1988.* **62 pages**

KSL 88-15
Tony Confrey and Fancois Daube; **GS2D: A 2D Geometry Systems,** March 1988. **15 pages**

KSL 88-16
Mark A. Musen; **Conceptual Models of Interactive Knowledge-Acquisition Tools,** March 1988. **9 pages**

KSL 88-19
Barbara Hayes-Roth, Micheal Hewett, M. Vaughan Johnson, and Alan Garvey; **ACCORD: A Framework for a Class of Design Tasks,** March 1988. **12 pages**

KSL 88-20
Barbara Hayes-Roth, Rattikorn Hewett, and Adam Seiver; **Diagnostic Explanation using Generic Models,** March 1988. **10 pages**

KSL 88-21
Alan Garvey and Barbara Hayes-Roth; **An Empirical Analysis of Explicit vs. Implicit Control Architectures,** March 1988. **20 pages**

KSL 88-22
Micheal Hewett and Barbara Hayes-Roth; **Real-Time I/O in Knowledge-Based Systems,** March 1988. **10 pages**

KSL 88-23
Robert Schulman and Barbara Hayes-Roth; **Plan-Based Construction of Strategic Explanations,** March 1988. **13 pages**

KSL 88-24
James F. Brinkley; **The Potential for Intelligent Three-Dimensional Ultrasound,** March 1988. To appear in: *Chervenak, F.A., Isaacson, G., Campbell, S. (eds.), Textbook of Ultrasound in Obstetrics and Gynecology, 1988.* **28 pages**

KSL 88-25
**(Working Paper)** Greg Byrd; **Modelling a Bus-Based Multiprocessor Using the CARE Simulation System,** March 1988. **11 pages**

KSL 88-26
Mark A. Musen; **Generation of Knowledge-Acquisition Tools from Clinical-Trial Models,** March 1988. To appear in: *Proceedings of Medical Informatics, Europe 1988. Oslo, Norway, August 1988.* **6 pages**

KSL 88-27
**(Working Paper)** H. J. Suermondt and Gregory F. Cooper; **Updating Probabilities in Multiply Connected Belief Networks,** March 1988. **9 pages**

KSL 88-28
Cooper, G.F.; **A method for using belief networks as influence diagrams,** April 1988.

KSL 88-34
**(Thesis)** Isabelle de Zegher-Geets; **IDEFIX: Intelligent Summarization of a Time-Oriented Medical Database,** June 1987. **99 pages**

KSL 88-38
Heckerman, D.E.; **An empirical comparison of three scoring schemes,** May 1988.

KSL 88-TBA
Chavez, R.M. and Cooper, G.F.; **KNET: Integrating hypermedia and normative Bayesian modeling.**

KSL 88-TBA
Lehmann, H., **Knowledge acquisition for probability-based expert systems.**

**Other Outside Articles:**

Cooper, G.F., Expert systems based on belief networks -- Current research directions, *Journal of Applied Statistical Models and Data Analysis*, **4**, 1988.

Cooper, G.F., invited commentary on: Lauritzen, S. and Spiegelhalter, D., Local computations with probabilities on graphical structures and their application to expert systems, *Journal of the Royal Statistical Society*, **B**, **50**, 1988.

Horvitz, E.J., Reasoning under varying and uncertain resource limitations, in: *Proceedings of the National Conference on Artificial Intelligence*, Minneapolis, MN, August, 1988.

Horvitz, E.J., Breese, J.S., and Henrion, M., Decision theory in expert systems and artificial intelligence, *Journal of Approximate Reasoning*, 1988.

## III.A.2.6. Resource Equipment

The SUMEX-AIM resource is a complex, integrated facility comprised of mainframes, workstations, networks, and servers illustrated in Figures 3 - 9. A key role of the SUMEX-AIM resource is to continue to evaluate workstations as the technology is changing rapidly. This evaluation includes new hardware and software, 1) to provide superior development and execution platforms for AI research, and 2) to support the ancillary "office environment" (presently carried out on the DEC 2060, which is being phased out). Thus far no single workstation has materialized that provides all the services we would like to see in support of either or both of these missions. This means that for the foreseeable future, we will utilize a multiplicity of machines and software to address the needs of the projects.

Systems based on the Motorola 68020 chip (e.g., SUN Microsystems or Apple Macintosh II workstations), the Intel 80286 and 80387 chips (e.g., IBM PS/1-4 machines), and other newer architectures such as reduced instruction set computer (RISC) chips, have Lisp benchmark data rivaling the performance of existing, specially microcoded Lisp machines (see Appendix B). It is still early to predict how this "race" will ultimately turn out and software environments play an equally important role with raw hardware speed in the decision. For now, the Lisp software environments on the "stock" machines are not nearly so extensively developed as on Lisp machines and conversely, the routine computing environments of Lisp machines (text processing, mail, spreadsheets, etc.) lag the tools available on stock UNIX machines.

In earlier year's we experimentally tried increasing usage of TI and Xerox Lisp machines (purchased as AI research platforms) for text editing, electronic mail, and document formatting with considerable success (although many of these tools were only tested in a prototype form and were not widely distributed). In addition, the use of expensive Lisp machines for routine computing and office applications impacts their availability as research tools.

We had been seeking an integration of both the Lisp machine and stock machine worlds. As discussed extensively in the progress section on Core Systems Research (see Page 39), these two capabilities came together as never before with the Macintosh II and microExplorer coprocessor systems.

### 1 - Purchases This Past Year

The SUMEX 2060 hardware continues to be stable and the relatively small amount of SUMEX-AIM money for new purchases has been concentrated on experimental workstations and server equipment needed for distributed system development. These purchases are paced carefully with the developments of higher performing, more compact, and lower cost systems. The NIH-funded purchases this past year are summarized below. For the most part, these represent evaluation units used to review the suitability of particular types of equipment preparatory to a larger volume purchase from non-NIH funding sources, as detailed earlier starting on Page 39).

1. Apple Mac-II
2. Rodime Disk Drive 100 Megabyte Disk Drive (for MAC-II)
3. E-Machines Big Picture Display (for MAC-II)
4. U.S. Robotics 9600 Baud Modems (2 ea).
5. Develcon X.25/Ethernet Gateway (shared cost with others)

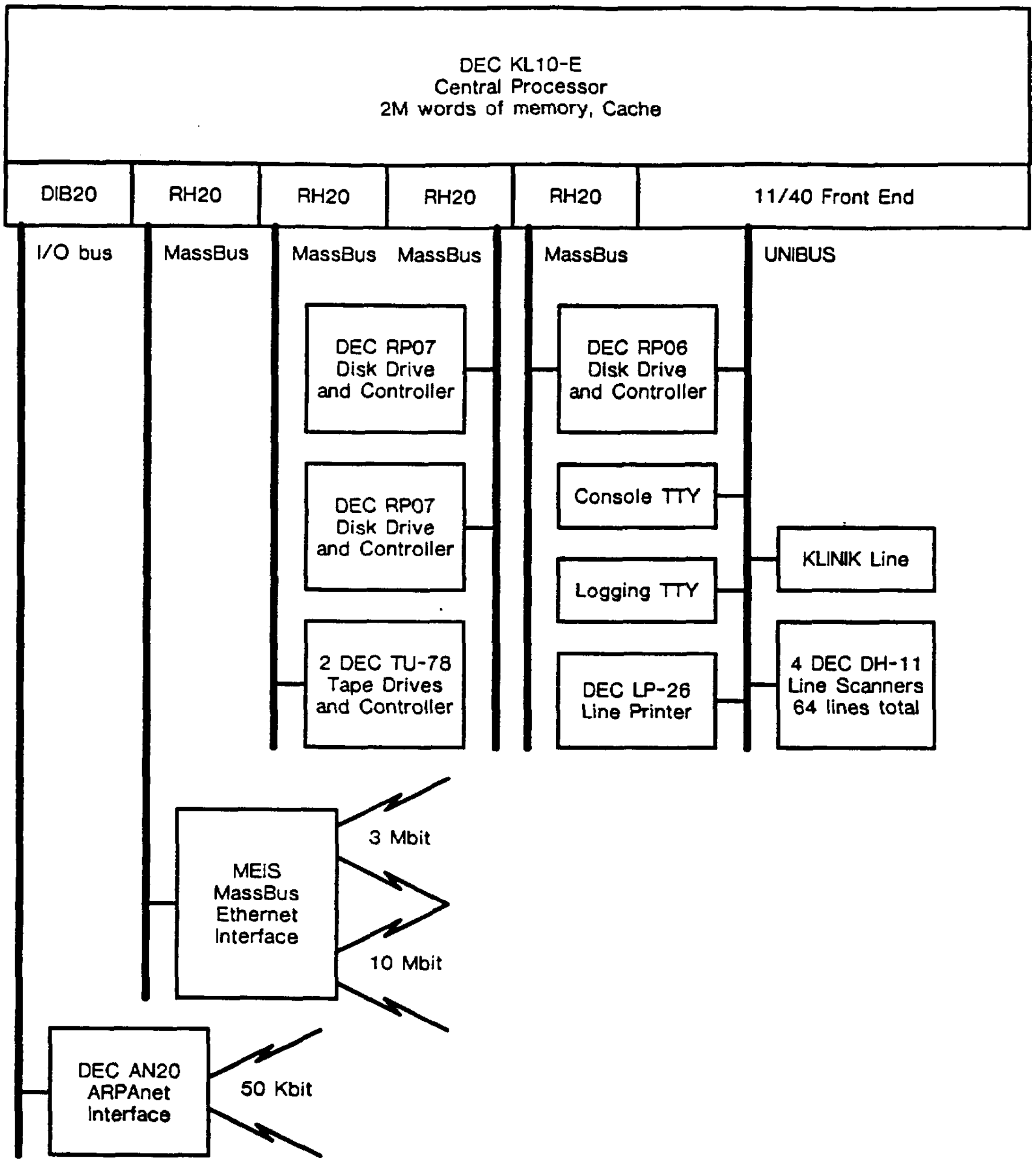


**Figure 3:** SUMEX-AIM DEC 2060 Configuration

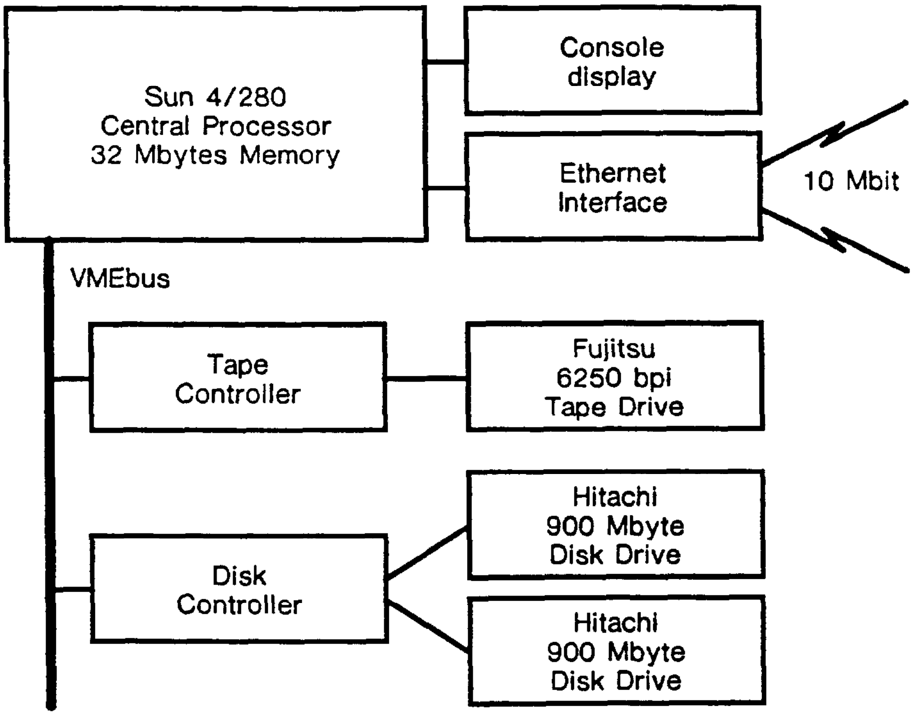


**Figure 4:** SUMEX-AIM SUN-4 Configuration

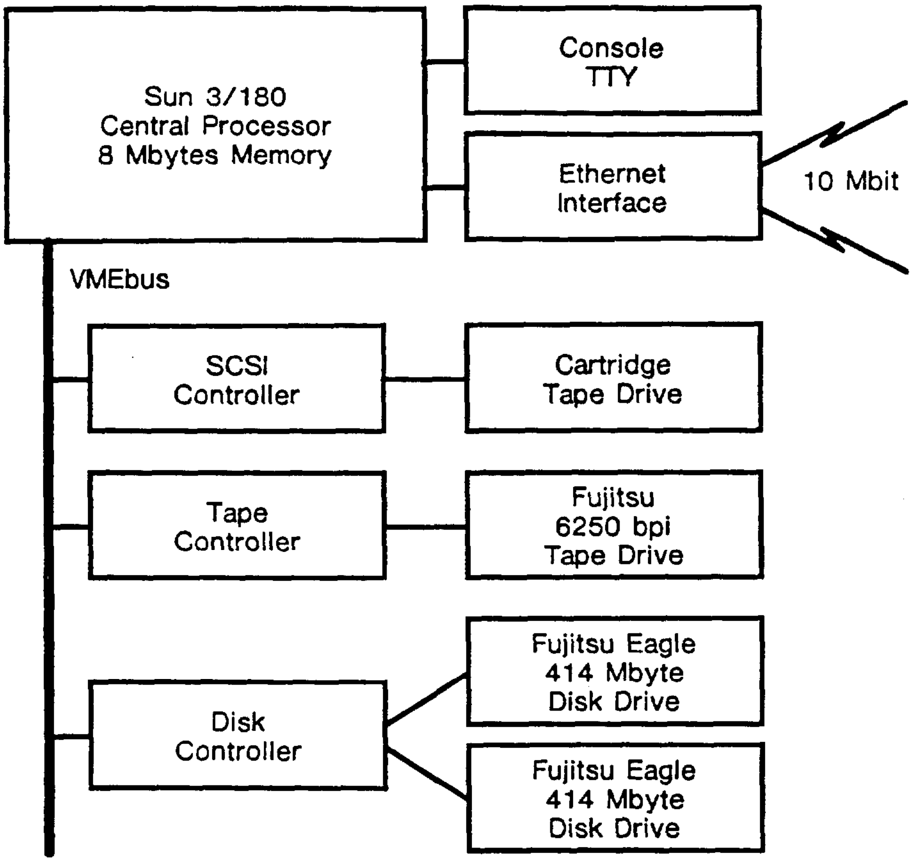


Figure 5: SUMEX-AIM Sun-3 File Server Configuration

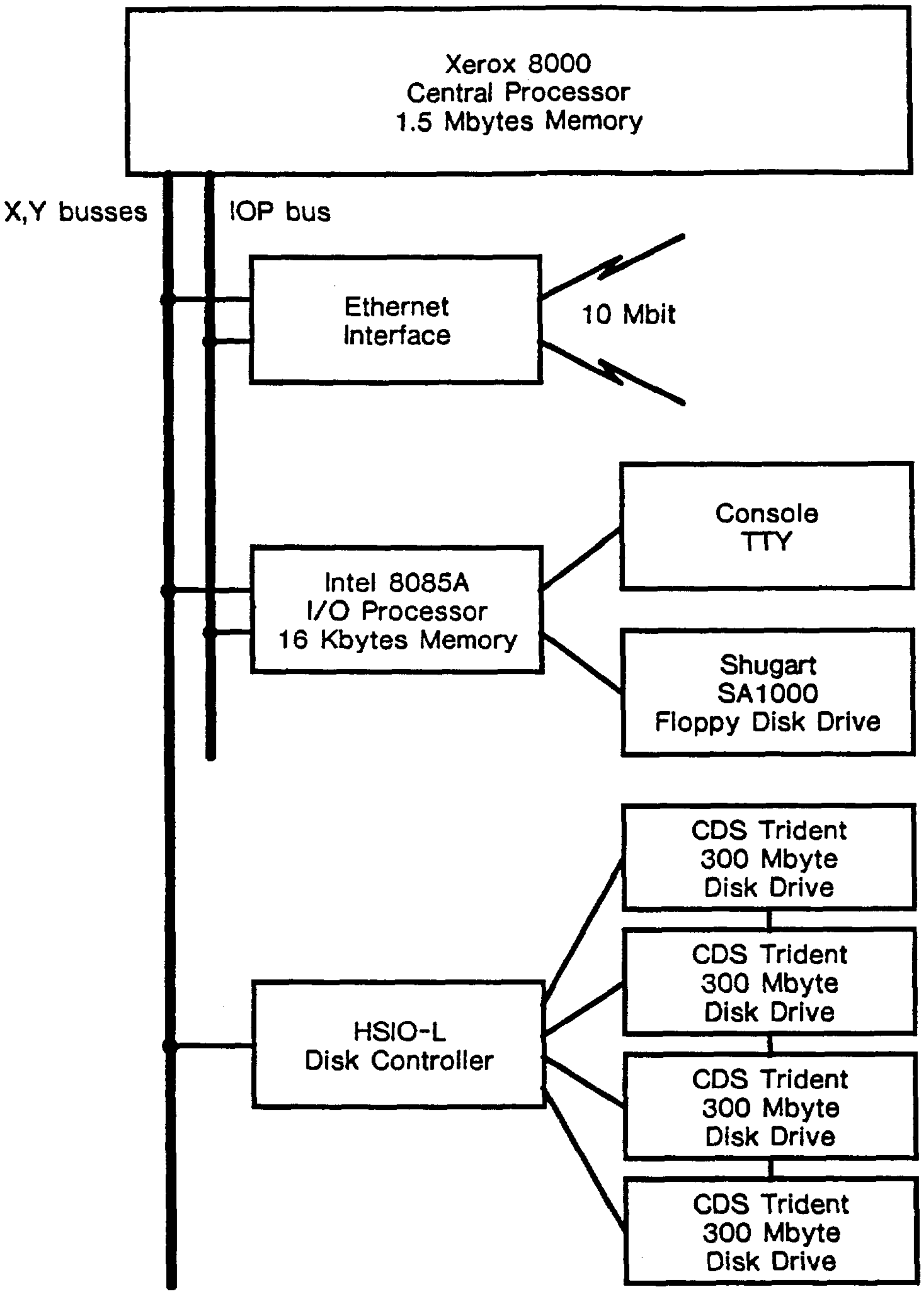


**Figure 6:** SUMEX-AIM Xerox File Server Configuration

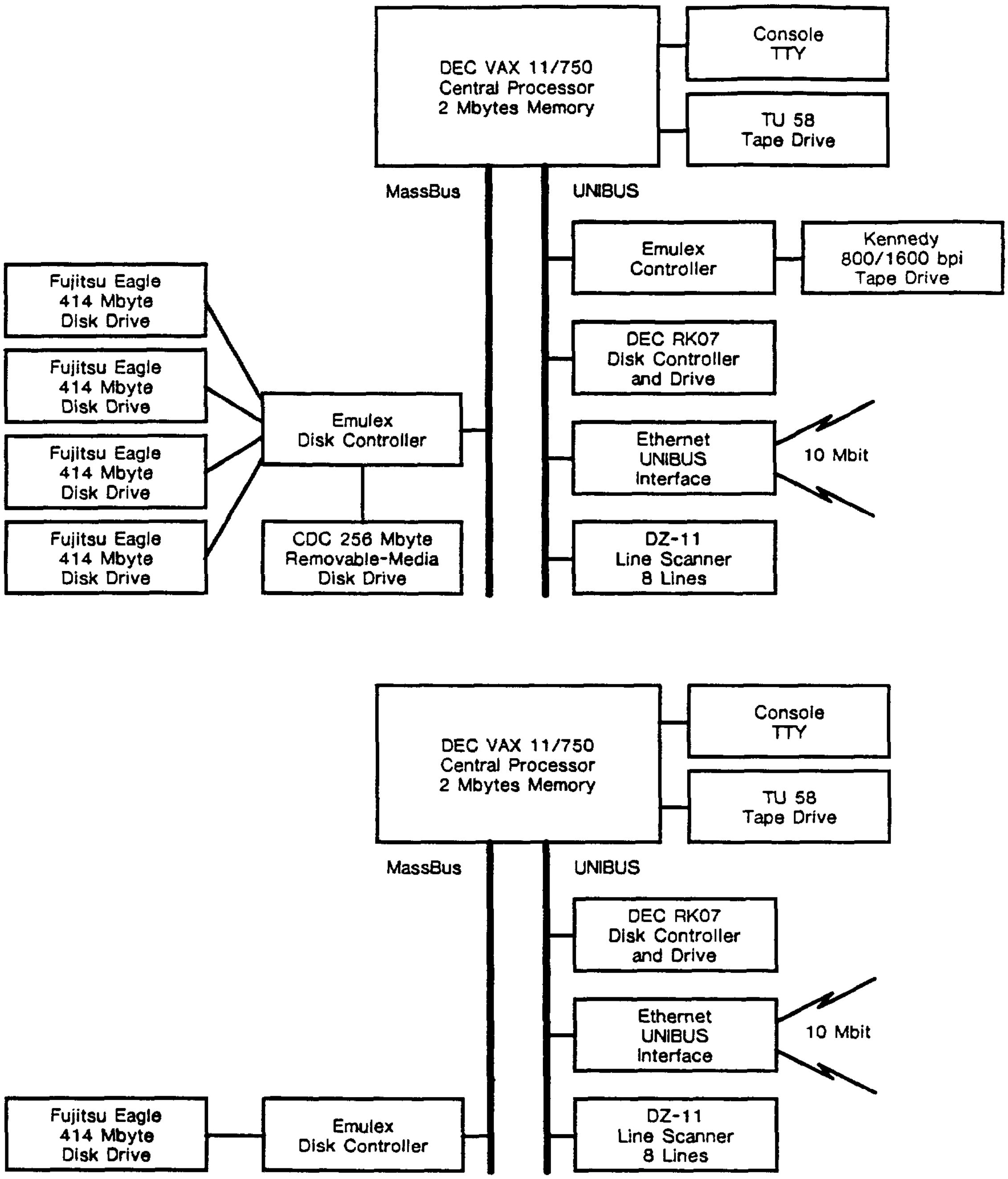


**Figure 7:** SUMEX-AIM VAX File Server Configuration

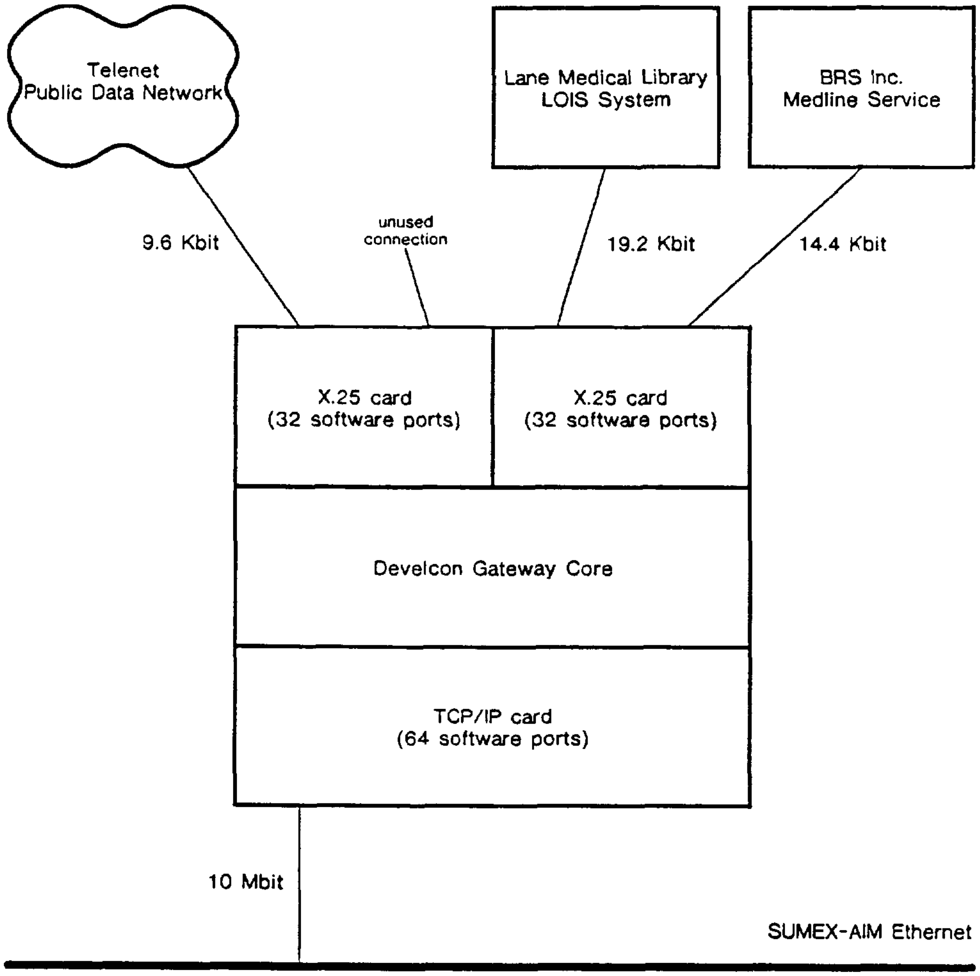


Figure 8: SUMEX-AIM Develcon X.25/TCP-IP Gateway Configuration

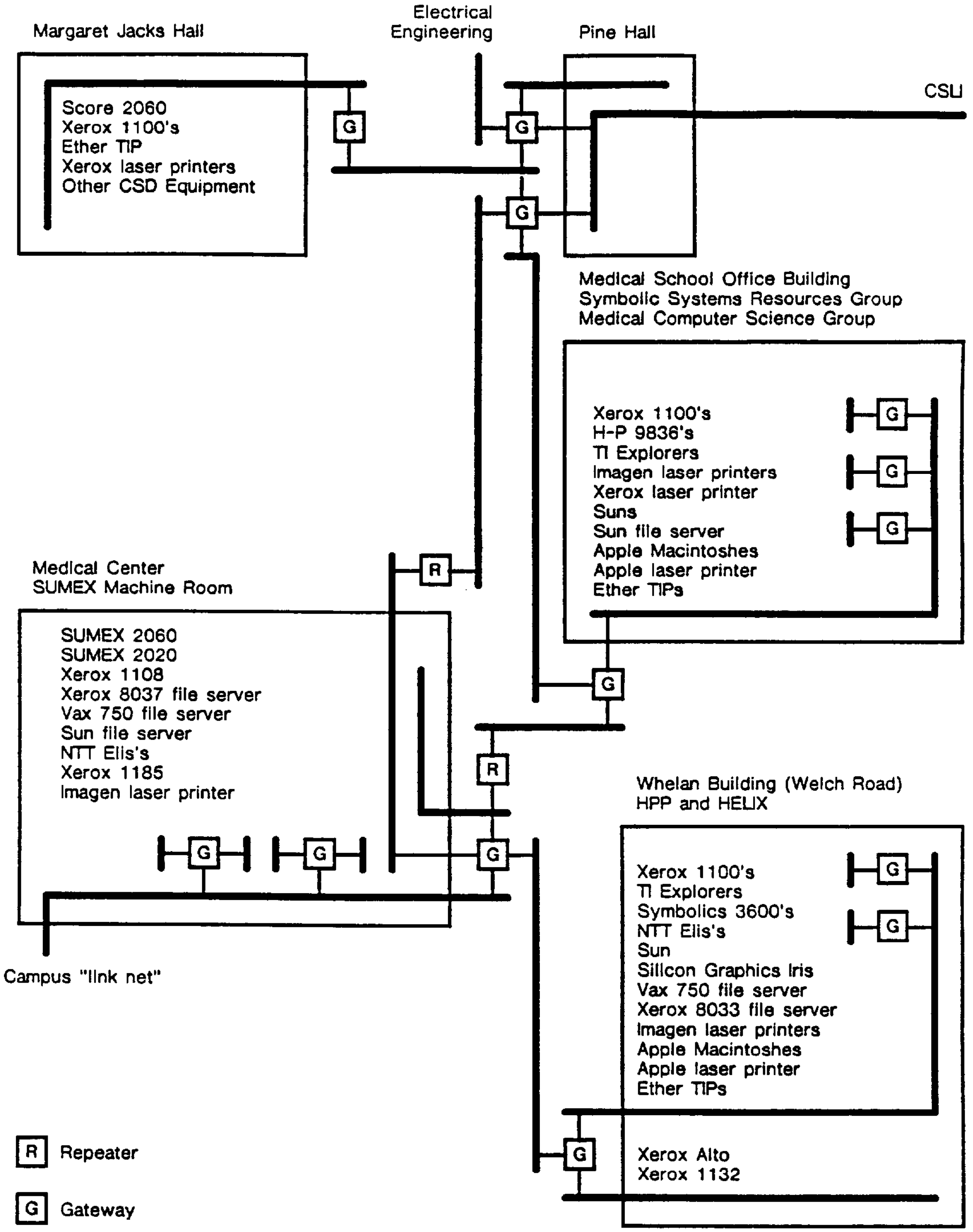


Figure 9: SUMEX-AIM Ethernet Configuration

## III.A.2.7. Training Activities

The SUMEX resource exists to facilitate biomedical artificial intelligence applications. This user orientation on the part of the facility and staff has been a unique feature of our resource and is responsible in large part for our success in community building. The resource staff has spent significant effort in assisting users to gain access to the SUMEX-AIM resources at Stanford and use it effectively as well as in assisting AIM projects in designing their own local computing resources based on SUMEX experience. We have also spent substantial effort to develop, maintain, and facilitate access to documentation and interactive help facilities. The HELP and Bulletin Board subsystems have been important in this effort to help users get familiar with the computing environment.

We have regularly accepted a number of scientific visitors for periods of several months to a year, to work with us to learn the techniques of expert system definition and building and to collaborate with us on specific projects. Our ability to accommodate such visitors is severely limited by space, computing, and manpower resources to support such visitors within the demands of our on-going research.

A tutorial was held in January 1988 on the Parallel Computing Architectures Project multiprocessor simulation system (see Page 29). This two-day session was attended by representatives from the DoD, NASA and Boeing, and included descriptions of the CARE/SIMPLE system, as well as the LAMINA programming interface. The attendees received instruction in use of the system for making measurements of the performance of various simulated multiprocessor applications.

Finally, the training of graduate students is an essential part of the research and educational activities of the KSL. Based largely on the SUMEX-AIM community environment, we have initiated two unique, special academic degree programs at Stanford, the Medical Information Science program and the Masters of Science in AI, to increase the number of students we produce for research and industry. A number of students are pursuing interdisciplinary programs and come from the Departments of Engineering, Mathematics, Education, and Medicine.

The *Medical Information Sciences (MIS)* program continues to be one of the most obvious signs of the local academic impact of the SUMEX-AIM resource. The MIS program received recent University approval (in October 1982) as an innovative training program that offers MS and PhD degrees to individuals with a career commitment to applying computers and decision sciences in the field of medicine. In Spring 1987, a University-appointed review group unanimously recommended that the degree program be continued for another five years. The MIS training program is based in the School of Medicine, directed by Dr. Shortliffe, co-directed by Dr. Fagan, and overseen by a group of six University faculty that includes two faculty from the Knowledge Systems Laboratory (Profs. Shortliffe and Buchanan). The specialized curriculum offered by the new program is intended to overcome the limitations of previous training options. It focuses on the development of a new generation of researchers with a commitment to developing new knowledge about optimal methods for developing practical computer-based solutions to biomedical needs.

The program accepted its first class of four trainees in the summer of 1983 and has now reached its steady-state size of approximately twenty-two graduate students. The program encourages applications from any of the following:

- medical students who wish to combine MD training with formal degree work and research experience in MIS;

- physicians who wish to obtain formal MIS training after their MD or their residency, perhaps in conjunction with a clinical fellowship at Stanford Medical Center;
- recent BA or BS graduates who have decided on a career applying computer science in the medical world;
- current Stanford undergraduates who wish to extend their Stanford training an extra year in order to obtain a "co-terminus" MS in the MIS program;
- recent PhD graduates who wish post-doctoral training, perhaps with the formal MS credential, to complement their primary field of training.

In addition, a special one-year MS program is available for established academic medical researchers who may wish to augment their computing and statistical skills during a sabbatical break. As of Spring 1988, 55% of our trainees have previously received MD degrees and another 23% are medical students enrolled in joint degree programs. 27% are candidates for the MS degree, while the rest are doctoral students. The program has seven graduates to date, with several more expecting to complete degrees before the end of 1988.

Except for the special one-year MS mentioned above, all students spend a minimum of two years at Stanford (four years for PhD students) and are expected to undertake significant research projects for either degree. Research opportunities abound, however, and they of course include the several Stanford AIM projects as well as research in psychological and formal statistical approaches to medical decision making, applied instrumentation, large medical databases, and a variety of other applications projects at the medical center and on the main campus. Several students are already contributing in major ways to the AIM projects and core research described elsewhere in this annual report.

We are pleased that the program already has an excellent reputation and is attracting superb candidates for training positions. The program's visibility and reputation is due to a number of factors:

- high quality students, many of whom publish their work in conference proceedings and refereed journals even before receiving their degrees; Stanford MIS students have won first prize in the student paper competition at the Symposium on Computer Applications in Medical Care (SCAMC) in 1985 and 1986, and have also received awards for their work at annual meetings of organizations such as the Society for Medical Decision Making, the American Association for Medical Systems and Informatics (AAMSI), and the American Association for Artificial Intelligence (AAAI);
- a rigorous curriculum that includes newly-developed course offerings that are available to the University's medical students, undergraduates, and computer science students as well as to the program's trainees;
- excellent computing facilities combined with ample and diverse opportunities for medical computer science and medical decision science research;
- the program's great potential for a beneficial impact upon health care delivery in the highly technologic but cost-sensitive era that lies ahead.

The program has been successful in raising financial and equipment support from industry and foundations. It is also recipient of a training grant from the National Library of Medicine. The latter grant was recently renewed for another five years with a study section review that praised both the training and the positive contribution of the SUMEX-AIM environment.

## III.A.2.8. Resource Operations and Usage

### 1 - Operations and Support

The diverse computing environment that SUMEX-AIM provides requires a significant effort at operations and support to keep the resource responsive to community project needs. This includes the planning and management of physical facilities such as machine rooms and communications, system operations routine to backup and retrieve user files in a timely manner, and user support for communications, systems, and software advice.

We spend significant time on new product review and evaluation such as Lisp workstations, terminals, communications equipment, network equipment, microprocessor systems, mainframe developments, and peripheral equipment. We also pay close attention to available video production and projection equipment, which has proved so useful in our dissemination efforts involving video tapes of our work.

We continue to operate the primary elements of our equipment base in a generally unattended manner. Operations costs are kept to a minimum by utilizing a student staff for routine tasks. Senior members of this staff provide improvements to the operations procedures in addition to training and supervising new students. This has provided SUMEX with a cost effective operations scheme, contributed to the education of the students, and assisted students in meeting their obligations in undergraduate financial aid programs.

While most of our equipment is concentrated in three computer equipment rooms, our move towards distributed computing has resulted in a substantial amount of equipment being installed in offices and student carrels. The planning of our project's area during the construction of the Medical School Office Building (described in last year's report) has made this distribution of equipment easier.

### 2 - Resource Usage Details

The following data give an overview of various aspects of SUMEX-AIM central resource usage. There are 5 subsections containing data respectively for:

1. Overall resource loading data (page 101).
2. Relative system loading by community (page 102).
3. Individual project and community usage (page 105).
4. Network usage data (page 111).
5. System reliability data (page 113).

For the most part, the data used for these plots cover the entire span of the SUMEX-AIM project. This includes data from both the KI-TENEX system and the current DECsystem 2060. At the point where the SUMEX-AIM community switched over to the 2060 (February, 1983), you will notice sharp changes in most of the graphs. This is due to differences in scheduling, accounting, and processor speed calculations between the systems.

*2.1 - Overall Resource Loading Data*

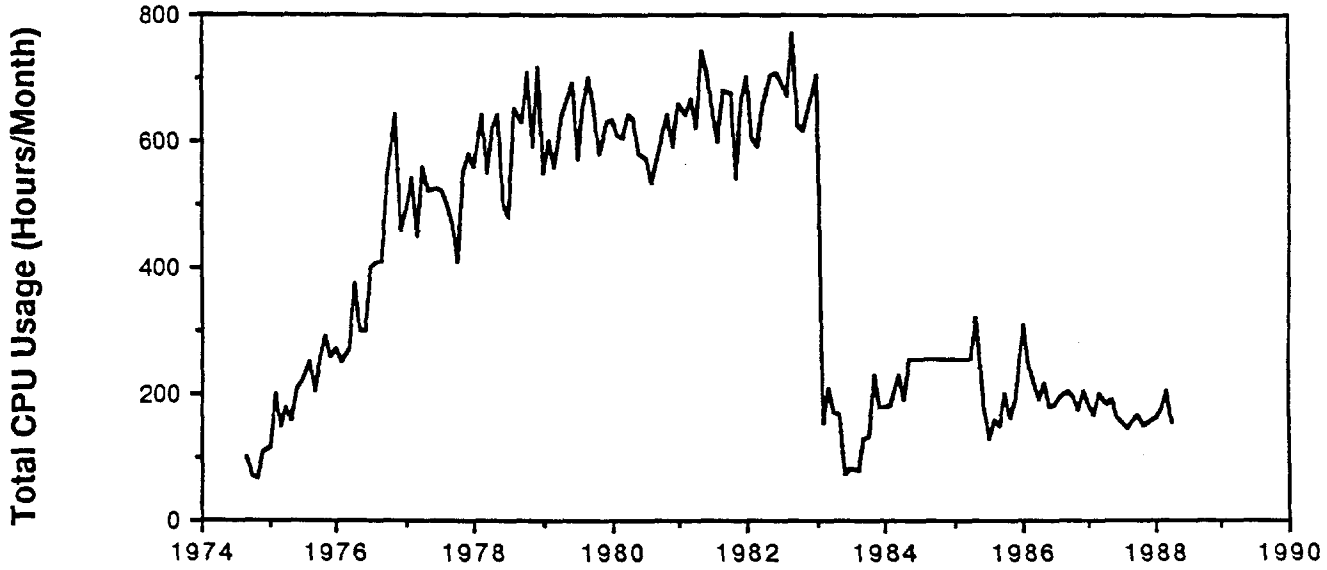


Figure 10: Total CPU Hours Consumed by Month

*2.2 - Relative System Loading by Community*

The SUMEX resource is divided, for administrative purposes, into three major communities: core ONCOCIN research, core AI research, and user projects based at the Stanford Medical School (*Stanford Projects*); user projects based outside of Stanford (*National AIM Projects*); and core system development efforts (*System Staff*). The initial resource management plan approved by the BRTP at the start of SUMEX specified that available system CPU capacity and file space resources were to be divided nominally between these communities in a 40:40:20 ratio. The "available" resources are those remaining after various monitor and community-wide functions (e.g., job scheduling, system overhead, network service, file space for subsystems, documentation, etc.) are accounted for.

The monthly usage of CPU resources and terminal connect time for each of these three communities is shown in the plots in Figure 11 and Figure 12. Many of the national user projects have already moved to their own machines for intensive research computing and now use SUMEX mainly for communications and information access. Hence, one might expect the proportion of CPU and file space use by Stanford projects, as compared to non-Stanford groups, to continue to grow correspondingly, as has been the case in the past. However, this past year there has been a dramatic increase in the national use of SUMEX-AIM for remote communications and information access. Much of this has been through the "anonymous" file transfer mechanism for which we cannot identify the user by name. We will attempt to record more information about such information access connections in future years.

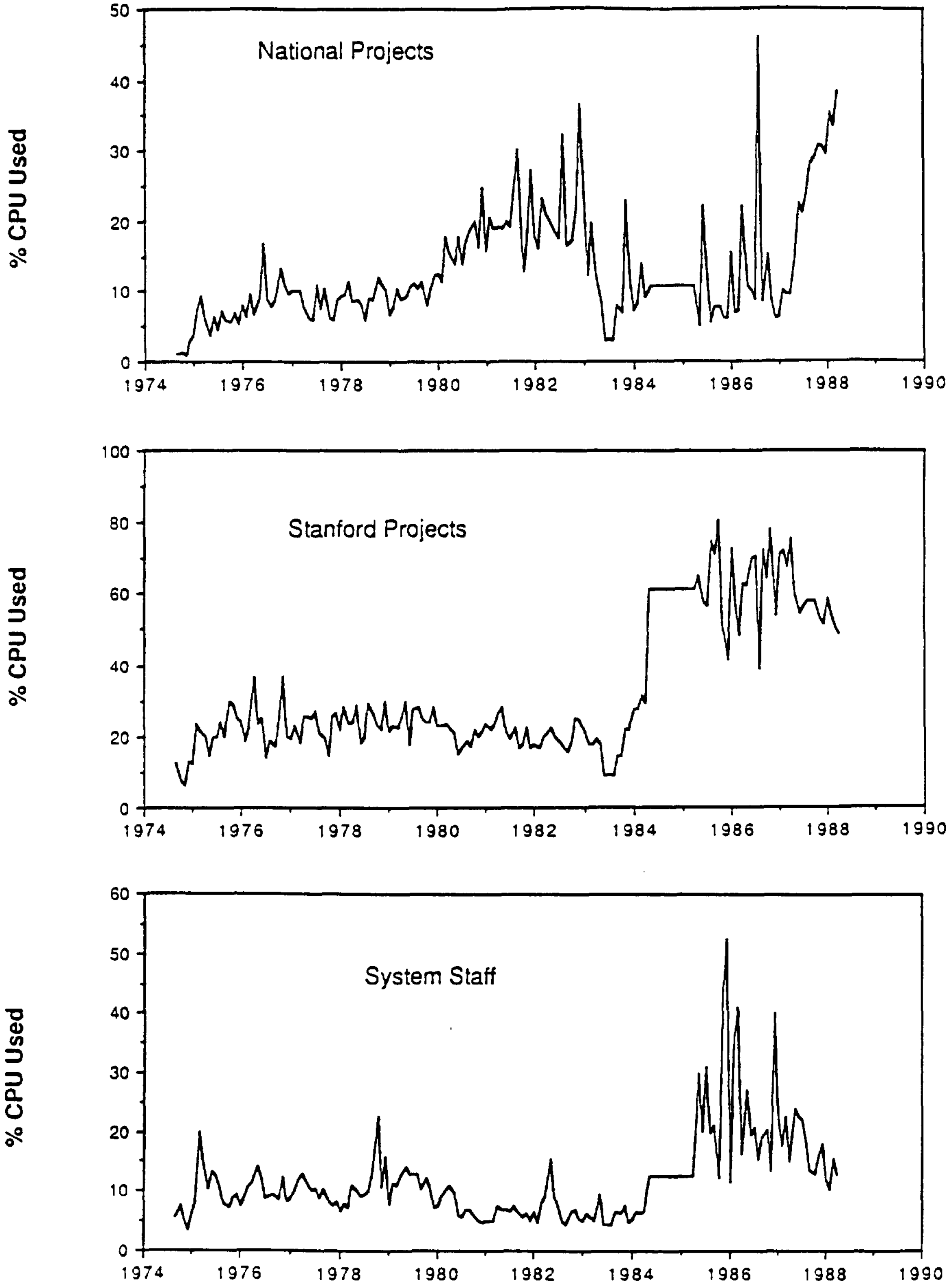


**Figure 11:** Monthly CPU Usage by Community

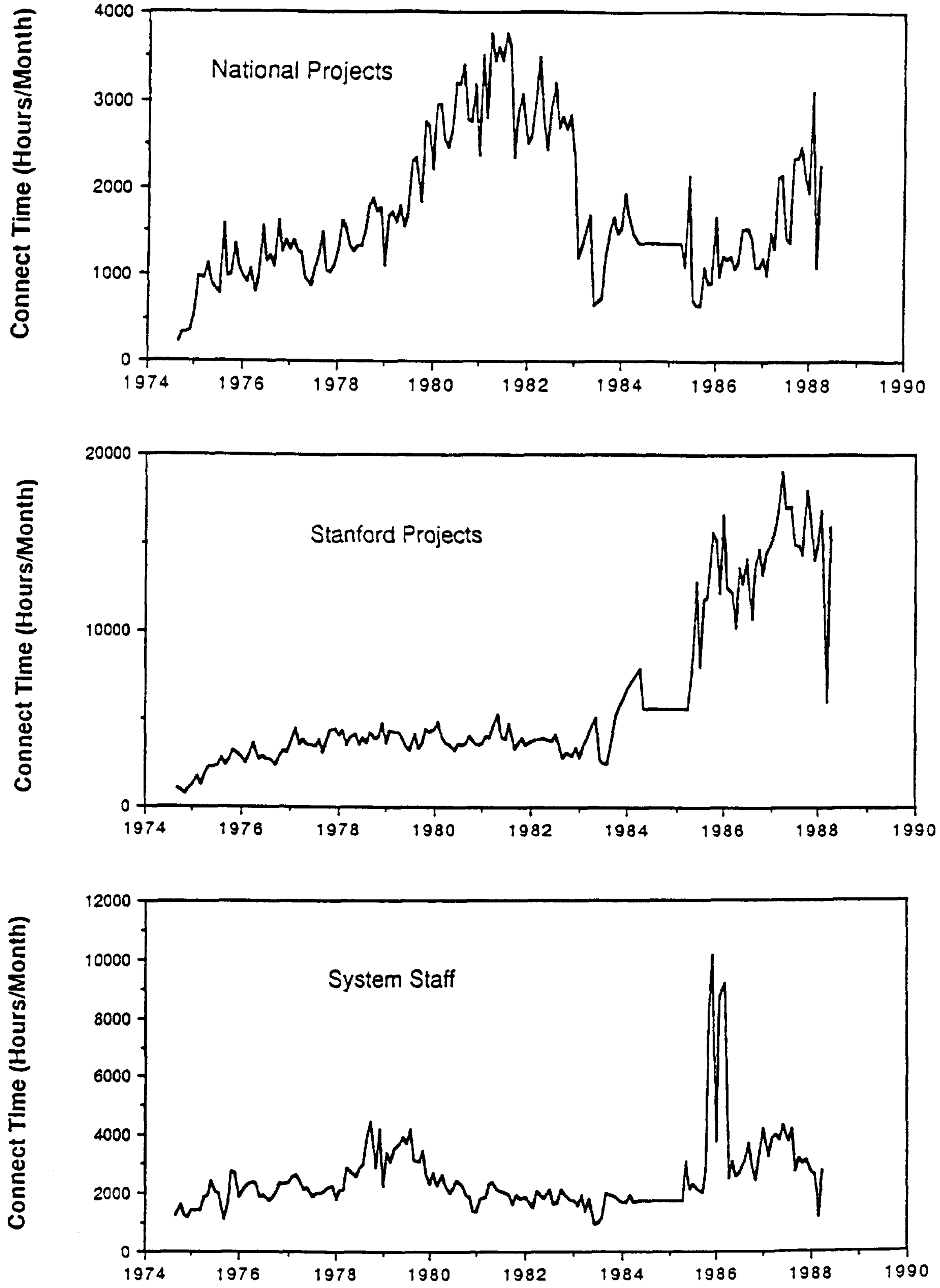


**Figure 12:** Monthly Terminal Connect Time by Community

### *2.3 - Individual Project and Community Usage*

The following histogram and table show cumulative resource usage by collaborative project and community during the past grant year. The histogram displays the project distribution of the total CPU time consumed between May 1, 1987 and April 30, 1988, on the SUMEX-AIM DECsystem 2060 system. Data include total CPU consumption by project (Hours), total terminal connect time by project (Hours), and average file space in use by project (Pages, 1 page = 512 computer words). These data were accumulated for each project for the months between May 1986 and April 1987.

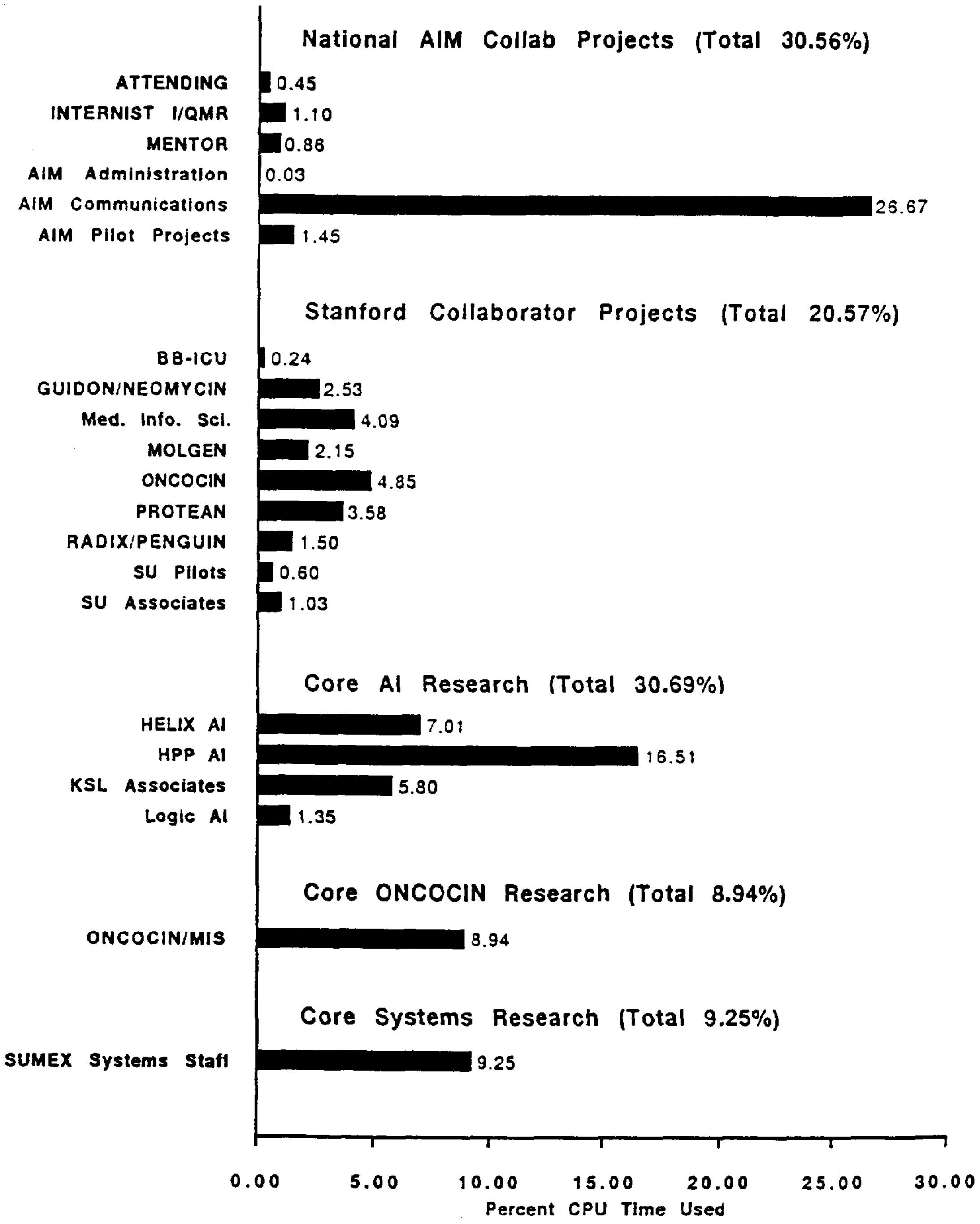


Figure 13: Cumulative CPU Usage Histogram by Project and Community

**Resource Use by Individual Project - 5/87 through 4/88**

| *National AIM Collaborator Community* | CPU (Hours) | Connect (Hours) | File Space (Pages) |
|---|---|---|---|
| 1) ATTENDING<br>"A Critiquing Approach to Expert Computer Advice"<br>Perry L. Miller, M.D., Ph.D.<br>Yale University School of Medicine | 5.08 | 217.81 | 650 |
| 2) INTERNIST-QMR Project<br>"Clinical Decision Systems Research Resource"<br>Jack D. Myers, M.D.<br>Randolph A. Miller, M.D.<br>University of Pittsburgh | 12.37 | 259.59 | 4693 |
| 3) MENTOR Project<br>"Medical Evaluation of Therapeutic Orders"<br>Stuart M. Speedie, Ph.D.<br>University of Maryland<br>Terrence F. Blaschke, M.D.<br>Stanford University | 9.62 | 4830.97 | 2000 |
| 4) AIM Pilot Projects | | | |
| PathFinder (Nathwani and Fagan) | 7.04 | 1002.52 | 1560 |
| Dynamic Systems (Widman) | 9.19 | 224.45 | 1433 |
| Radiation Therapy (Kalet) | 0.02 | 1.42 | 4 |
| 5) AIM Communications | | | |
| AIM Mail-Only Users | 6.64 | 970.79 | 3567 |
| AAAI Management | 5.50 | 1930.65 | 929 |
| BIONET | 2.32 | 168.76 | 680 |
| MCS Collaborators | 7.25 | 1796.62 | 1110 |
| MOLGEN Collaborators | 2.99 | 310.63 | 883 |
| Anonymous File/ Information Access | 273.22 | 13593.75 | 233579 |
| Other | 0.62 | 32.77 | 839 |
| 6) AIM Administration | 0.32 | 44.53 | 2009 |
| | --------- | ---------- | ------- |
| Community Totals | 342.18 | 25385.27 | 253936 |

**Figure 14:** Table of Resource Use by Project

| *Stanford Collaborator Community* | CPU (Hours) | Connect (Hours) | File Space (Pages) |
|---|---|---|---|
| 1) BBICU Project<br>Lawrence M. Fagan, M.D., Ph.D.<br>Department of Medicine<br>Barbara Hayes-Roth, Ph.D.<br>Computer Science Department | 2.65 | 1512.22 | 237 |
| 2) GUIDON-NEOMYCIN Project<br>William J. Clancey, Ph.D.<br>Bruce G. Buchanan, Ph.D.<br>Dept. Computer Science | 28.37 | 6359.48 | 1862 |
| 3) Medical Information Sciences<br>Edward H. Shortliffe, M.D., Ph.D.<br>Lawrence M. Fagan, M.D., Ph.D.<br>Department of Medicine | 45.76 | 11821.74 | 4128 |
| 4) MOLGEN Project<br>"Applications of Artificial Intelligence to Molecular Biology: Research in Theory Formation, Testing and Modification"<br>Edward A. Feigenbaum, Ph.D.<br>Peter Friedland, Ph.D.<br>Charles Yanofsky, Ph.D.<br>Depts. Computer Science/Biology | 24.04 | 6523.52 | 7114 |
| 5) ONCOCIN Project<br>"Knowledge Engineering for Medical Consultation"<br>Edward H. Shortliffe, M.D., Ph.D.<br>Lawrence M. Fagan, M.D., Ph.D.<br>Department of Medicine | 54.31 | 11759.34 | 7940 |
| 6) PROTEAN Project<br>Oleg Jardetzky<br>School of Medicine<br>Bruce Buchanan<br>Computer Science Department | 40.04 | 8795.76 | 4097 |
| 7) RADIX-PENGUIN Project<br>Gio C.M. Wiederhold, Ph.D.<br>Depts. Computer Science/ Medicine | 16.84 | 3150.80 | 9284 |

Figure 14: Table of Resource Use by Project, Continued

| | | | |
|---|---|---|---|
| 8) Stanford Pilot Projects<br>REFEREE Project (Buchanan) | 6.72 | 1380.48 | 452 |
| 9) Stanford Associates | 11.55 | 8064.25 | 3265 |
| | --------- | ---------- | ------- |
| Community Totals | 230.29 | 59367.60 | 38380 |

| | CPU | Connect | File Space |
|---|---|---|---|
| *Core AI Research* | (Hours) | (Hours) | (Pages) |
| 1) ABLE Project<br>Robert S. Engelmore, Ph.D.<br>Computer Science Department<br>Scott Clearwater, Ph.D.<br>Los Alamos National Laboratory | 11.85 | 5372.31 | 570 |
| 2) Advanced Architectures<br>Edward A. Feigenbaum, Ph.D.<br>Computer Science Department | 105.23 | 32008.01 | 7284 |
| 3) Blackboard Architectures<br>Barbara Hayes-Roth, Ph.D.<br>Computer Science Department | 53.33 | 10390.84 | 4636 |
| 4) DART Project<br>Michael R. Genesereth, Ph.D.<br>Computer Science Department | 6.10 | 3220.27 | 0 |
| 5) Financial Resource Management<br>Bruce G. Buchanan, Ph.D.<br>Thomas C. Rindfleisch<br>Computer Science Department | 26.07 | 7449.24 | 2180 |
| 6) Intelligent Agents Project<br>Michael R. Genesereth, Ph.D.<br>Computer Science Department | 6.32 | 507.92 | 0 |
| 7) Knowledge Engineering Studies<br>Bruce G. Buchanan, Ph.D.<br>Dianna Forsythe, Ph.D.<br>Computer Science Department | 5.66 | 1313.59 | 236 |
| 8) Machine Learning Studies<br>Bruce G. Buchanan, Ph.D.<br>Computer Science Department | 34.23 | 9774.46 | 6134 |

Figure 14: Table of Resource Use by Project, Continued

| | | | |
|---|---|---|---|
| 9) MRS Project<br>Michael R. Genesereth, Ph.D.<br>Computer Science Department | 2.75 | 1336.60 | 0 |
| 10) SOAR Project<br>Paul R. Rosenbloom, Ph.D.<br>Information Sciences Institute<br>University of Southern California | 12.57 | 6979.07 | 907 |
| 11) Software Design Project<br>H. Penny Nii<br>Computer Science Department | 1.13 | 124.99 | 904 |
| 12) Very Large Knowledge Bases<br>Edward A. Feigenbaum, Ph.D.<br>Richard Keller, Ph.D.<br>Computer Science Department | 13.33 | 3153.20 | 1298 |
| 13) KSL Administration | 55.15 | 13910.04 | 10090 |
| 14) KSL Associates | 9.84 | 1645.64 | 1554 |
| | --------- | ---------- | ------- |
| Community Totals | 343.56 | 97186.17 | 35793 |

| *Core ONCOCIN Research* | CPU<br>(Hours) | Connect<br>(Hours) | File Space<br>(Pages) |
|---|---|---|---|
| 1) Core ONCOCIN and Medical Information Sciences<br>Edward H. Shortliffe, M.D., Ph.D.<br>Lawrence M. Fagan, M.D., Ph.D.<br>Department of Medicine | 100.07 | 23581.08 | 12068 |
| | --------- | ---------- | ------- |
| Community Totals | 100.07 | 23581.08 | 12068 |

| *Core Systems Research* | CPU<br>(Hours) | Connect<br>(Hours) | File Space<br>(Pages) |
|---|---|---|---|
| 1) SUMEX Staff R & D<br>Thomas C. Rindfleisch<br>Departments of Medicine and Computer Science | 98.27 | 18547.71 | 10008 |

Figure 14: Table of Resource Use by Project, Continued

| | | | |
|---|---|---|---|
| 2) System Associates | 5.24 | 252.56 | 1201 |
| Community Totals | 103.51 | 18800.29 | 11209 |

| *System Operations* | CPU (Hours) | Connect (Hours) | File Space (Pages) |
|---|---|---|---|
| 1) System Operations | 650.01 | 80316.52 | 3437 |
| 2) SUMEX Staff Opns & Mgmnt | 98.27 | 18547.71 | 10008 |
| Community Totals | 748.28 | 98864.23 | 13445 |
| Resource Grand Totals | 1867.89 | 323184.64 | 364831 |

Figure 14: Table of Resource Use by Project, Concluded

### *2.4 - Network Usage Statistics*

The plots in Figures 15 and 16 show the monthly network terminal connect time for the public data networks and the INTERNET usage. The INTERNET is a broader term for what was previously referred to as ARPANET usage. Since many vendors now support the INTERNET protocols (TCP/IP) in addition to the ARPANET, which converted to TCP/IP in January of 1983, it is no longer possible to distinguish between ARPANET usage and Internet usage on our 2060 system. Similarly, after we switched to the Develcon gateway between the TELENET X.25 network and our TCP/IP Ethernet, we are not able to distinguish TELENET 2060 users from other Internet users. We are hoping to refine the accounting services available from the Develcon gateway so we will have a separate log of connection activity.

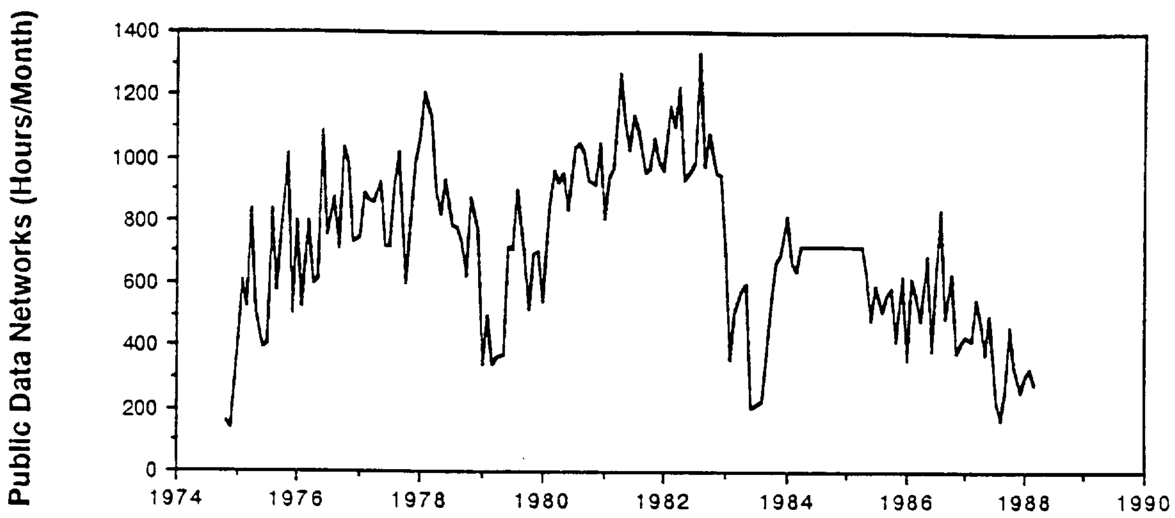


Figure 15: Public Data Network Terminal Connect Time

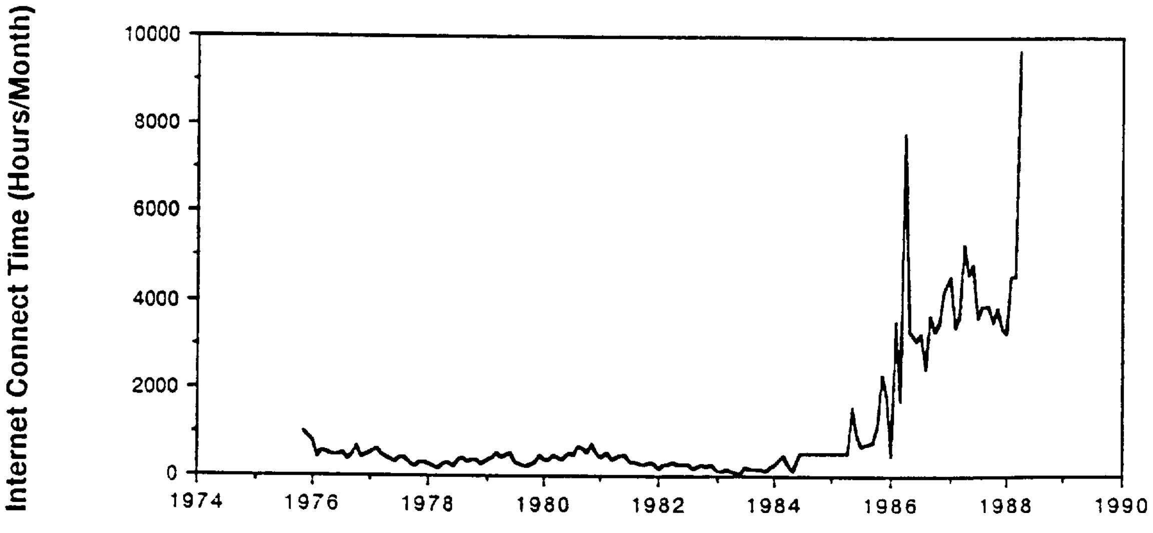


Figure 16: INTERNET Terminal Connect Time

### *2.5 - System Reliability*

As in past years, the reliability of the 2060 system has been very high. The data below cover both hardware- and software-related system failures. The time listed under preventive maintenance (PM) downtime includes both the time required for hardware PM and the time required for installation of updated system software. The data cover the period of May 1, 1987 to April 30, 1988.

May 1987 - April 1988:

| May | Jun | Jul | Aug | Sep | Oct | Nov | Dec | Jan | Feb | Mar | Apr |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 14.2 | 3.2 | 5.0 | 0.1 | 24.0 | 10.9 | 2.6 | 4.0 | 5.4 | 2.4 | 35.2 | 2.2 |

**Figure 17:** 2060 Downtime Summary -- Hours per Month

May 1987 - April 1988:

| | |
|---|---|
| Reporting period: | 366 days, 0 hours, 12 min |
| Total Uptime: | 361 days, 10 hours, 56 min |
| Uptime Percentage: | 98.99% |
| PM Downtime: | 0 days, 20 hours, 9 min |
| Actual Downtime: | 3 days, 17 hours, 6 min |
| Total Downtime: | 4 days, 13 hours, 16 min |
| MTBF:[1] | 5 days, 15 hours, 32 min |

**Figure 18:** Overall 2060 Reliability Summary

[1] Mean Time Between Failures

## III.B. Highlights

In this section we describe several research highlights from the past year's activities. These include notes on existing projects that have passed important milestones, new pilot projects that have shown progress in their initial stages, and other core research and special activities that reflect the progress, impact, and influence the SUMEX-AIM resource has had in the scientific and educational communities.

## III.B.1. PROTEGE -- Developing Knowledge Acquisition Tools for Clinical Trial Advice Systems

Knowledge acquisition, the process whereby computer scientists (*knowledge engineers*) interview experts in a given application area and attempt to encode the experts' specialized knowledge in a computer program, is widely recognized as a principal obstacle in the development of knowledge-based systems. To ease these difficulties, workers in medical artificial intelligence have experimented with a number of computer-based tools designed to facilitate the construction of clinical advice systems. One such tool is OPAL, a program that allows physicians to enter descriptions of oncology treatment plans directly into the knowledge base of ONCOCIN, an expert system that offers advice concerning protocol-directed cancer therapy. Physicians who use OPAL do not have to understand the production rules and other data structures that are used internally by ONCOCIN to represent cancer-therapy knowledge. Rather, oncologists enter knowledge into OPAL by drawing flowcharts and by filling in the "blanks" of graphical forms that anticipate the concepts required to define cancer therapy protocols. OPAL automatically converts the physicians' specifications into the knowledge representations that ONCOCIN uses to generate its treatment advice.

In 1986, system builders entered 36 oncology protocols into the ONCOCIN knowledge base using OPAL. Encoding these protocols via traditional knowledge engineering techniques might well have taken several person-years. Although OPAL clearly streamlines knowledge entry for the ONCOCIN system, OPAL itself required about 3.5 person-years of software development before it was ready for routine use. Furthermore, because OPAL takes advantage of a detailed model of therapy planning in oncology, the program is very much domain-dependent. OPAL is of no use, for example, to endocrinologists who desire to create knowledge bases for therapy planning for thyroid disease or to cardiologists interested in treating heart failure.

PROTEGE is a system that allows expert system developers to create knowledge acquisition tools that are much like OPAL (i.e., have convenient and powerful user interfaces), but that are custom tailored for new application areas. PROTEGE permits knowledge engineers to define models of the *kinds of clinical trials* that occur in various areas of medicine. It then uses these models to produce domain-specific knowledge acquisition tools that allow physicians to define new protocols by filling in graphical forms and by drawing flowchart diagrams. To date, PROTEGE has been used to create two such tools: 1) p-OPAL -- a program that incorporates most of the functionality of OPAL and thus acquires knowledge concerning clinical trials in oncology and 2) HTN -- a knowledge acquisition tool for hypertension drug studies. Producing each of these knowledge acquisition tools required a knowledge engineer using PROTEGE to enter *models* for the relevant classes of clinical trials (oncology and hypertension, respectively), defining those clinical trial models in terms of a general model of treatment planning built into the PROTEGE system itself. The PROTEGE user fills in the blanks of various forms to define models for given types of clinical trial applications, much as the user of OPAL fills out graphical forms to define individual cancer protocols. The forms in PROTEGE directly reflect the general model of treatment planning. Once the clinical trial models had been specified, PROTEGE generates the corresponding knowledge acquisition tools (computer programs) automatically.

## III.B.2. A Speech Interface to ONCOCIN

Motivated by a recurrent request from collaborating clinicians to augment the "classical" keyboard interface to expert systems with a speech-based interface and by recent technological advances in continuous-speech systems, we began a project to explore the integration of speech input with the ONCOCIN cancer therapy advisor system. The project uses a commercial continuous-speech system loaned to us by Speech Systems, Inc. (SSI) of Tarzana, California. The speech recognizer consists of a custom microelectronic processor and a suite of special speech decoding software modules.

This experiment has taken advantage of our on-going work in distributed computing since the phonetic device, initial parsing software, and the ONCOCIN system all run on different pieces of hardware. Researchers have developed a prototype network connection and command interpreter between the speech module (running on a SUN workstation) and the Xerox 1186 computer that runs ONCOCIN and have designed a series of modifications to the ONCOCIN user interface has in turn been modified to accept verbal commands.

The prototype interface system permits users to navigate the graphical ONCOCIN interface and enter clinical data using speech. The system uses the location of the cursor on the screen to provide a context for choosing candidate grammars with which to attempt to recognize the user's utterance. The system dynamically adjusts the list of candidate recognition grammars based on the on-going dialog and it is now possible to carry on most of the ONCOCIN data acquisition steps using speech alone or speech plus pointing with the mouse. In addition, some input data elements (such as the neural toxicities) can be entered as textual descriptions and automatically encoded on the 1-4 point numerical scale used on oncology flowsheet forms.

We are also performing experiments to enhance the system's grammars with a wider range of phrases clinicians actually use when talking to a computer and to gain insights into clinicians' models of spoken interaction with advice systems. This will allow us to ground our interface design better in observed practice. In order to assess how physicians would speak to a computer in an ideal situation, we are simulating fully functional continuous-speech understanding with a hidden computer operator generating the output of ONCOCIN as if it had the ability to understand all spoken input. A video camera records both audio and visual clues. The physicians use ONCOCIN in the same manner as it is used in the clinic when they see patients, but with the added capability of (simulated) speech input. These experiments enable us to build up both a basic vocabulary for the real speech system as well as examine subtle linguistic issues to guide future directions.

## III.B.3. SIMPLE/CARE -- Emulation of Parallel Computing Architectures

Many applications require knowledge-based systems that can cope with large amounts of data and produce responses in real-time. The current hardware and software architectures for knowledge-based systems cannot support such requirements. The most promising approach for achieving orders of magnitude improvement in the quantitative performance of knowledge-based systems is by exploiting concurrency on multiprocessor systems. Based on projections for integrated circuit technologies, it is clear that highly parallel multiprocessor computers, consisting of 100's to 1000's of processors and realizing a variety of concurrent architectures, can be built. A major computer science issue is whether such computers can be used effectively to enhance the performance of knowledge-based systems. Since 1985, the Knowledge Systems Laboratory at Stanford University has been investigating these issues.

The goals and technical approach of this project, largely supported by DARPA under the Strategic Computing Program, have been to achieve two to three orders of magnitude speed-up in the execution of knowledge-based systems, by identifying and exploiting sources of concurrency at all levels of system design: the application level, the problem-solving framework level, the programming language level and the hardware systems architecture level. Due to the inherent complexity of the task and the lack of theoretical foundations for parallel computation with ill-structured problems, we have taken an empirical approach.

Simulation of systems at an architectural level offers an effective way to study critical design choices. SIMPLE/CARE is a powerful simulation system that forms the foundation for our empirical investigations. SIMPLE is a CAD (Computer Aided Design) system for hierarchical, multiple-level specification of computer architectures and includes an associated mixed-mode, event-based simulator. CARE is a parameterized, multiprocessor array emulation defined in SIMPLE's specification languages and running on SIMPLE's simulator. Our simulation system has been used internally to make quantitative comparisons of the performance of various architectures and to gain insights into how different concurrent programming models support the development of concurrent applications decomposition and organization. The system is in use by several research groups at Stanford and it has been exported to several other sites, including NASA Ames Research Center. A tutorial was held in January 1988, attended by representatives from the DoD, NASA and Boeing, which described the CARE/SIMPLE system. The attendees received instruction in use of the system for making measurements of the performance of various simulated multiprocessor applications. A Stanford graduate course on these tools is currently in progress this spring quarter.

## III.B.4. Toward the Distributed SUMEX-AIM Community

We have made a key decision this past year on the core system definition that will support the first phase of the distributed AIM community. Guided by our requirements to provide powerful and widely-available tools for general computing and biomedical research and to sharply focus our limited development resources on a small number of standardized hardware and software configurations, we considered a wide range of alternative systems for AIM community computing needs to replace and upgrade the services of the 2060. Based on dominant user preferences for the icon-based interface, outstanding technical performance, very competitive academic pricing, and an already-growing group of national AIM users, we have chosen Apple Macintosh II workstations as the general computing environment for researchers and staff, TI Explorer Lisp machines (including the microExplorer Macintosh coprocessor) as the near-term high-performance Lisp research environment, and a SUN-4 as the central system network server (network services, file services, printing services, etc.). To actually implement a prototype of the planned distributed environment, a substantial quantity of hardware was purchased with DARPA research funds in the spring of 1988. We are now in the midst of the installation and integration process, concentrating initially on getting basic capabilities operational, such as for text processing, filing/archiving, printing, graphics, office management, system building tools, information resource access, and distributed system operation and management.

Initial user response to the introduction of these systems has been overwhelmingly enthusiastic, even though there are many "rough edges" remaining to be smoothed out in the systems integration. Our core development work for the environment of the Mac II-, Explorer-, and SUN-based system has focussed on providing remote access between workstations themselves and with servers, integrating a solid support of the TCP-IP network protocol, and building a powerful distributed electronic mail system. The new Mac II mail system will be an adaptation of the prototype distributed system developed in recent years for the Xerox Lisp machine and which is in routine use by a number of people and is being ported to the Explorer.

One of the key issues in selecting the systems for our distributed computing environment was the performance of Common Lisp and to help make this evaluation, we undertook an informal survey of the performance of two KSL AI software packages, SOAR and BB1, on a wide variety of machines. Within a factor of two of the best performance, a considerable range of workstations based on stock microprocessor chips (e.g., the Motorola 680xx and the Intel 80386) as well as specially microprogrammed Lisp chips have comparable performance. Even though performance gaps between microprogrammed Lisp systems and stock workstation implementations are narrowing, there still remains a significant difference in the quality of the development environments. We have attempted to distill and promote the commercial development of the key features of the Lisp machine environments that would be needed in stock machine implementations in order to make them attractive in a development setting.

After the prototype distributed system is implemented and tested in the Stanford KSL environment, we will package and document its elements so that other sites in the AIM community and beyond can duplicate its capabilities. As this work progresses we will phase out the old DEC 2060 to be replaced by the SUN-4 as a general community communication and information server.

## III.C. Administrative Changes

There have been few administrative changes within the project this past reporting year. Professor Shortliffe had been on sabbatical at the University of Pennsylvania the year before last and returned to Stanford in mid-July 1987, when he resumed his role as SUMEX Principal Investigator.

We continue to operate the cost recovery system we reported on last year as part of phasing out BRTP subsidy of the DEC 2060 facility. The details of this system are discussed on page 121. In summary, we are successfully recovering the projected 40% of 2060 operations costs this year ($136,374) from Stanford users, with the declining component of NIH support (60% this year) used to protect national users from fees for service, including communications. This additional burden on Stanford projects continues to be absorbed almost entirely in existing direct cost budgets since no supplements for new computing costs were forthcoming in the middle of on- going grant and contract awards. This has affected staffing and student support directly in our labor-intensive research efforts. All of our new support applications are being written with requests for funds to cover projected computing charges.

This next year we will increase the cost recovery goal to 60% of projected 2060 operations costs as scheduled in our grant application of June 1985. We also plan to physically phase out the 2060 and replace it with the new SUN-4 network server if technical development activities follow on schedule. The detailed interaction of this transition with our cost recovery procedures remain to be worked out during this coming year.

# III.D. Resource Management and Allocation

## III.D.1. Overall Management Plan

Early in the design of the SUMEX-AIM resource, an effective management plan was worked out with the Biotechnology Resources Program (now Biomedical Research Technology Program) at NIH to assure fair administration of the resource for both Stanford and national users and to provide a framework for recruitment and development of a scientifically meritorious community of application projects. This structure has been described in some detail in earlier reports and is documented in our recent renewal application. It has continued to function effectively as summarized below.

- The AIM Executive Committee meets periodically by teleconference to advise on new user applications, discuss resource management policies, plan workshop activities, and conduct other community business. The Advisory Group meets as needed to review project applications. (See Appendix D for a current listing of AIM committee membership).

- We actively recruit new application projects and disseminate information about AI in biomedicine. With the development of more decentralized computing resources within the·AIM community outside of Stanford, the use of SUMEX resources by AIM members has shifted more and more toward communication with colleagues and access to information.

- With the advice of the Executive Committee, we have opened SUMEX-AIM resources widely to biomedical users desiring electronic communications facilities. A list of current users who have used SUMEX for this purpose over this past year is given starting on Page IV.E.

- We have carefully reviewed on-going projects with our management committees to maintain a high scientific quality and relevance to our biomedical AI goals.

- We continue to provide active support for the AIM workshops. The most recent one was held this spring at Stanford University, under the auspices of the American Association for Artificial Intelligence (AAAI).

- We have continued to provide systems advice to users attempting to set up computing resources at their own sites, based on the expertise developed in the SUMEX resource environment.

- We have tailored resource policies to aid users whenever possible within our research mandate and available facilities.

## III.D.2. 2060 Cost Center

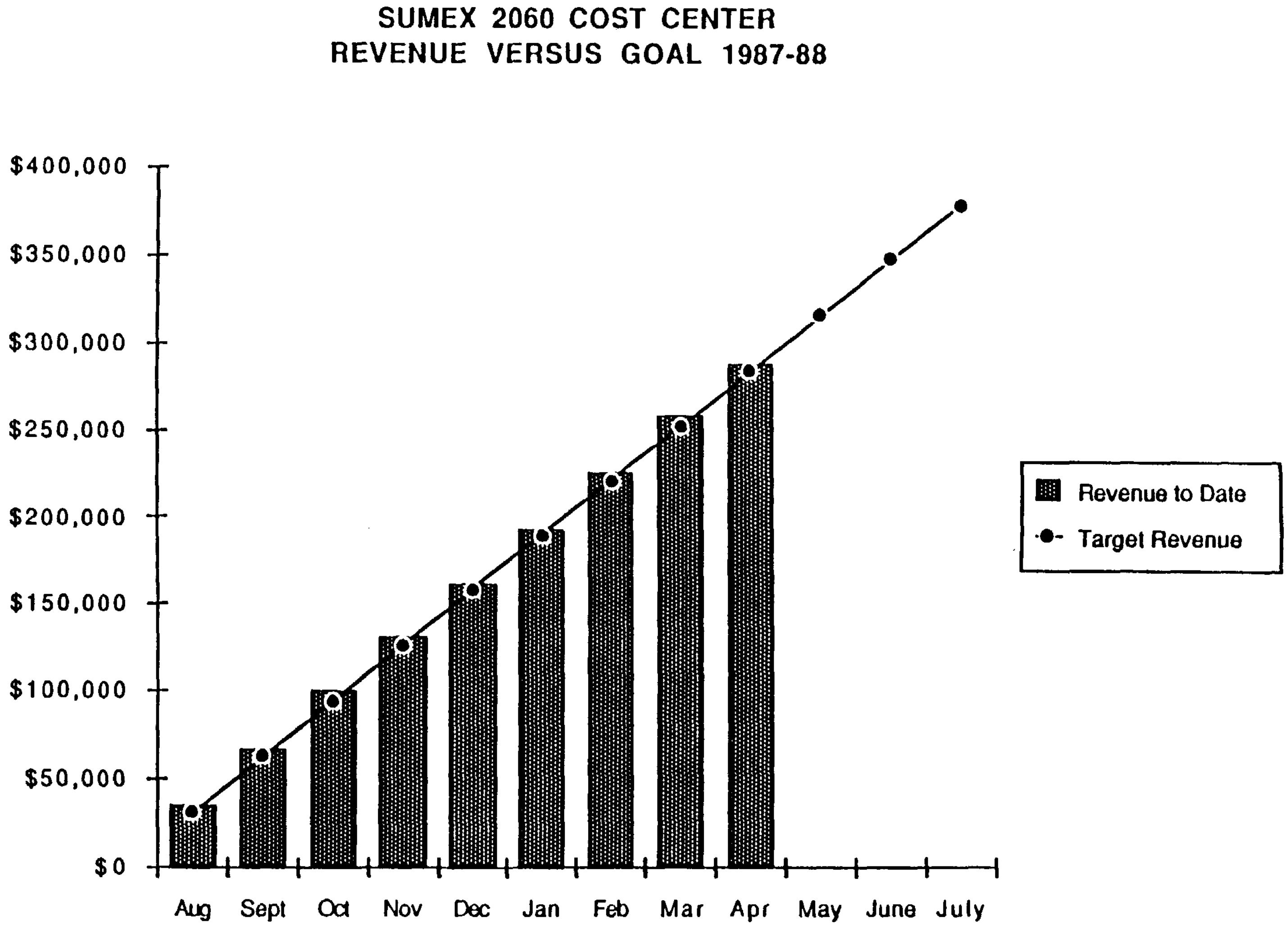
### General Cost Center Structure

Our plan for the term of the current grant is a firm but responsible transition of the SUMEX-AIM resource to a distributed community model of operation. There has continued to be a group of national and local users -- particularly young projects needing seed support prior to obtaining major funding -- that depend on a central shared resource like the SUMEX mainframe. In addition, the 2060 has played a key role as a central server for intercommunity communication and shared information. Powerful and widely available workstation equipment is rapidly becoming accessible, however, at a cost that most projects can afford, even young ones. Thus, the period of critical dependence on the DEC 2060 for raw computing cycles is largely past and its role in supporting routine computing and communication services can also soon be replaced by other more cost effective means. We are in the process of phasing out the SUMEX 2060 machine over the next few years in favor of the new distributed workstation environment we are developing. This process is progressing gradually and responsibly so that our users can relocate to other facilities or move to workstation environments for their research without disruption.

Specifically, our renewal proposal for the five-year period 8/1/86-7/31/91, submitted to the Division of Research Resources in June 1985, called for phasing out NIH support for DEC 2060 mainframe operations over the course of the grant period and the establishment of a cost center at Stanford to recover the unsubsidized costs of 2060 operations from the established Stanford user community. This phase-out process is taking place linearly over five years, with 80% of the 2060 costs charged to the resource budget in renewal year 1 (Grant Year 14), 60% in year 2 (current grant year), 40% in year 3, 20% in year 4, and 0% in year 5, when routine operations (even national user services) will be supported entirely by user revenues. Use of the 2060 by members of the national AIM community is still free of charge at this time, and we will continue to cover the total cost of national community 2060 usage from the NIH subsidy as long as funding permits.

In keeping with this plan, during the summer of 1986, we requested and received the approval of the Government Cost and Rate Studies section of Stanford's Controller's Office to establish a 2060 cost center effective at the start of renewal year 1 (August 1, 1986) with a charge per CPU hour based on our projections of 2060 operations costs and anticipated billable CPU usage.

In last year's annual report, we reported success in recovering the 20% of 2060 operations costs not subsidized by NIH from our Stanford user community during our first year of cost center operation. This year our objective was to recover 40% of 2060 costs from Stanford users with a corresponding increase in the rate charged per CPU hour as of August 1, 1987, the start of renewal year 2. We have been monitoring cost center expenses and revenues carefully this second year and again anticipate breaking even at the end of the cost center's fiscal year at the end of July. Figure 19 shows the cumulative user revenues collected by month for the period August 1987 through April 1988.



REVENUE AS OF 4/30/88: $289,211
TARGET REVENUE FOR YEAR: $379,988

Figure 19: 2060 Cost Center Performance

## III.E. Dissemination of Resource Information

We are continuing our past practice of making a substantial effort to disseminate the AI technology developed here. This has taken the form of many publications -- over forty-five combined books and papers are published per year by the KSL; wide distribution of our software, including systems software and AI application and tool software, both to other research laboratories and for commercial development; production of films and video tapes depicting aspects of our work; and significant project efforts at studying the dissemination of individual applications systems such as the ONCOCIN resource-related research project (see 144).

### Software Distribution

We have widely distributed both our system software and our AI tool software. Since much of our general system-level software is distributed via the ARPANET we do not have complete records of the extent of the distribution. Software such as TOPS-20 monitor enhancements, the Ethernet gateway and TIP programs, the SEAGATE AppleNet to Ethernet gateway, the PUP Leaf server, the SUMACC development system for Macintosh workstations, and our Lisp workstation programs are frequently distributed in this manner to the ARPANET community and beyond.

Our primary distribution effort is directed towards the AI tools we have developed. In recent years, the volume of inquiries for this type of software and requests for tapes has been a substantial burden on the staff and so it was decided to turn over most of this type of software distribution to Stanford's Office of Technology Licensing (OTL). This organization handles software distribution and technology licensing matters for much of the Stanford community. Since there are several OTL staff members assigned to the distribution of Stanford software, requests for information and tapes are handled quickly and efficiently. Also, OTL's staff has the expertise needed to handle the legal questions that frequently arise in the distribution of software, and an established computerized record-keeping scheme. SUMEX staff continues to be available as needed to assist OTL with special administrative and technical matters.

Specific software distribution events this past year include:

- The Parallel Computing Architectures Project multiprocessor simulation system, CARE/SIMPLE, is in use by several research groups at Stanford and it has been ported to several external sites including NASA Ames Research Center.

- Two (2) licenses were granted for the EMYCIN package and twenty-two (22) licenses were granted for the BB1 package.

- OTL has concluded a license arrangement with Cisco, Inc. for the commercial development and marketing of the SUMEX Ethernet gateway and TIP service software.

- The agreement between Stanford University and Kinetics Inc. covering hardware and software technology for an Ethernet-to-AppleTalk gateway has been converted from an on-going royalty agreement to a fully paid license.

- OTL reported the expiration of an exclusive license to Molecular Designs Ltd. covering some aspects of the software generated by the DENDRAL Project. The source code and binary versions of this software are now available to all users (commercial, government, and academic) through OTL.

### AIM Community Systems Support

We continue to make a special effort to assist other members of the SUMEX-AIM community in integrating the technologies needed for biomedical AI research. This is often achieved through direct contact with staff members at these institutions, (e.g., with Professor Sticklen's group at Michigan State University after his move from Ohio State and with Professor Widman's group at the University of Texas), at meetings and workshops, or via electronic mailing lists. For example, the Info-MAC, Info-Explorer, and Info-1100 mailing lists have hundreds of members and cover a broad range of equipment issues, software issues, and topics in artificial intelligence.

### Video Tapes and Films

The KSL has continued to prepare video tapes that provide an overview of the research and research methodologies underlying our work and that demonstrate the capabilities of particular systems. These tapes are available through our groups, the Fleischmann Learning Center at the Stanford Medical Center, and the Stanford Computer Forum, and copies have been mailed to program offices of our various funding sponsors. In addition to the earlier tapes covering Knowledge Engineering in the KSL, ONCOCIN Overview, and ONCOCIN Demonstration, we have recent tapes on the PROTEAN project, the BB1 project, and a one-day symposium on KSL research activities.

# III.F. Suggestions and Comments

### Resource Organization

We continue to believe that the Biomedical Research Technology 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 and merging of the appropriate technological and medical disciplines.

### Electronic Communications

SUMEX-AIM has pioneered in developing more effective methods for facilitating scientific communication. Whereas face-to-face contacts continue to play a key role, in the longer-term computer-based communications will become increasingly important to the NIH and the distributed resources of the biomedical community. We would like to see the BRTP take a more active role in promoting these tools within the NIH and its grantee community. This is particularly important in the light of significant on-going changes to the national networking environment (see Page 75).

## IV. Description of 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. In addition to these detailed progress reports, abstracts for each project and its individual users are submitted on a separate Scientific Subproject Form. However, we have included briefer summary abstracts of the fully-authorized projects in Appendix E on page 277.

Those groups from the National AIM community which use the SUMEX-AIM resource solely for communication (i.e., electronic mail to and from colleagues or access to bulletin boards and other information resources at SUMEX) are listed starting on Page 220, without detailed reports on their research.

The detailed 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

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
   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. The only exceptions are the respective lists of relevant publications which have been uniformly formatted for parallel reporting on the Scientific Subproject Form.

## IV.A. 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 Shortliffe as Principal Investigator.

In addition to the progress reports presented here, abstracts for each project and its individual users are submitted on a separate Scientific Subproject Form.

# IV.A.1. BBICU Project

## BBICU Project: Blackboard Applications in the Intensive Care Unit

Adam Seiver, M.D.
Department of Surgery
Veterans Administration Hospital
Palo Alto, California

Lawrence Fagan, M.D., Ph.D.
Department of Medicine
Stanford University

Barbara Hayes-Roth, Ph.D.
Department of Computer Science
Stanford University

## I. SUMMARY OF RESEARCH PROGRAM

### *A. Project Rationale*

We are designing a data-interpretation and therapy-planning system for the intensive care unit (ICU). Fundamental research issues in temporal reasoning are associated with the ICU application area including assimilation of incoming data, representation of time-oriented intervals, and description of ongoing physiological processes [Fagan 84]. In addition, in ICUs of the 1990s, many more physiological measurements will need to be collected at frequent intervals, and increased access to the current medical record in coded format will be possible. Processing of incoming data will have to be opportunistic, selecting from a number of models that have different computational requirements and accuracy. We will use a blackboard architecture, known as BB1, that has evolved from earlier work on protein-structure elucidation and construction layout [Hayes-Roth 85]. BB1 is particularly well suited for the ICU project because it maintains separate blackboards for domain and control knowledge.

Although we have selected the blackboard structure as the organizing principle, many knowledge representation issues remain. First, we must represent the structure and function (anatomy and physiology) of the respiratory system. By characterizing the pathophysiology in terms of generic faults, we will create a more flexible means to diagnose problems in unusual situations -- in contrast to the phenomenological rules used in earlier systems.

Second, we must coalesce quantitative and qualitative models. The physiology of the respiratory and cardiac systems have been modeled in detail, but it is impractical to base the entire reasoning process on complex mathematical equations. Instead, we are developing methods to transform quantitative models into simpler formulations. We must make explicit the simplifying assumptions and associate them with their corresponding models, so that we can select an applicable model for situations of varying complexity.

Using the approach of our project for planning treatments for cancer patients [Langlotz 87], we will use strategic knowledge to create patient-specific specializations of standard treatment plans. We will use decision analytic methods to evaluate and explain the various treatment options available at any point in time. The long-term goal of this project is to embed the decision-making components within the data management tasks of the ICU.

### *B. Medical Relevance and Collaboration*

The problem of too much data being generated in the ICU is well recognized. Originally, monitors were designed to provide more objective assessments of the physiology of the patients in life-threatening situations. However, as more and more measurements became available, the ability of clinicians to assimilate the data began to drop. Expert systems can be designed to sort through the data, recognize untoward events in context and help with therapy selection. An early version of this was Fagan's VM system, which was based on extensions to the production rule framework. The current research has far broader goals, including real-time response using multiple methods for reasoning, reasoning from anatomy and physiology, and the use of integrated mathematical models. This has led to a three way collaboration between the VA hospital which is installing a data management system for a new Surgical ICU (Seiver), the Computer Science Department where blackboard research for real-time systems is in progress (Hayes-Roth), and the Medical Computer Science Group where investigations in qualitative-quantitative reasoning is taking place (Fagan).

### *C. Highlights of Research Progress*

This project is in its early stages, but we are beginning to see progress in both research directions. The BB1 group has demonstrated a system that can reason by analogy from a set of symptoms corresponding to a physical "blockage" back to generic knowledge about structures involved in a flow process. This system has been developed in the BB1 architecture on the Explorer Lisp machine. This group has developed a prototype that can:

1. monitor a few types of electronically sensed data in real time,

2. dynamically focus attention on different types of data depending on the current situation,

3. incrementally classify asynchronously arriving data into temporal episodes of normal/abnormal physiological parameters,

4. dynamically compute conditional probabilities of alternative diagnoses as newly arriving data are classified, and

5. create alternative hypotheses when compiled clinical knowledge fails to explain the situation by reformulating the problem in terms of an underlying model of the structure and function of the body.

To date, these goals have been carried out with a small number of data streams and recognized diagnoses. A major research goal of this project is to understand how the knowledge representations and processing techniques will have to adapt as the problem is scaled up.

The quantitative/qualitative group has concentrated on building a variety of mathematical models based on the Dickenson model of oxygen transport through the circulatory system. These equations are at different levels of specification, and heuristics are used to select the most appropriate model at each point in time. Another active area of research is the relationship between the therapy planning module and the solution of mathematical models used for prediction.

*D. Relevant Publications*

1. Fagan, L., Kunz, J., Feigenbaum, E, and Osborn, J. Adapting a rule-based system for a monitoring task, in *Rule Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project*, B. Buchanan and E. Shortliffe (eds.). Reading, MA: Addison-Wesley Publishing Co., 1984.

2. Hayes-Roth, B. A Blackboard architecture for control. (Artificial Intelligence) 26:251-321, 1985.

3. Langlotz, C., Fagan, L., Tu, S., Sikic, B., and Shortliffe, E. A therapy planning architecture that combines decision theory and artificial intelligence techniques. *Computers and Biomedical Research* 20:279-303, 1987.

## II. INTERACTIONS WITH THE SUMEX-AIM RESOURCE

*A. Medical Collaborations and Program Dissemination via SUMEX*

As described above, this project is a three-way collaboration between the Departments of Computer Science and Medicine, and Department of surgery at the VA Hospital. As, such we will need electronic mail and networking facilities. In addition, we imagine strong interactions with other projects around the world with similar research goals. We have already been contacted by research groups in Holland, Scotland, and Norway. In addition, similar research projects are underway at Yale, Berkeley, and Chicago. We expect that the networking facilities may allow us to share test cases, and possibly knowledge bases.

*B. Sharing and Interaction with Other SUMEX-AIM Projects*

The Yale project mentioned above is associated with Perry Miller's group. We also expect considerable interaction with the ONCOCIN and other parts of the Heuristic Programming Project at Stanford. The temporal issues involved with this project are relevant to Larry Widman. As the last AIM Workshop at Stanford in April, 1988, Larry demonstrated his programs on SSRG-based Explorers with the aid of SSRG staff member Rich Acuff.

*C. Critique of Resource Management*

The SUMEX staff have been quite helpful in the support of the various machines that have been used in this project so far. We anticipate that the project will migrate to Micro-Explorers and/or to Mac IIs. We believe that the current efforts of the SUMEX staff are quite appropriate for our research needs.

## III. RESEARCH PLANS

Our basic research agenda is described above. The basic research issues underlying this project will extend for several years, leading to an implementation in the Veterans Administration Hospital in Palo Alto.

This research will continue to need help assistance with local area networking, file service, and inter-network mail. We will need support for communications support within a project that is spread out over three geographical sites, and working with related but not identical hardware.

Since this is a new project, we are only depending on the DEC-20 for mail, file service, and communication facilities. The SUN-4 arrangement should be able to provide these services if local area network mail is fully implemented.

The SUMEX staff has been quite useful in providing support in other configurations of mainframe and workstations networked together. We anticipate that support for our unique collaborative arrangement will be equally superb.

# IV.A.2. GUIDON/NEOMYCIN Project

**GUIDON/NEOMYCIN Project**

**William J. Clancey, Ph.D.**
**Department Computer Science**
**Stanford University**

**Bruce G. Buchanan, Ph.D.**
**Computer Science Department**
**Stanford University**

## I. SUMMARY OF RESEARCH PROGRAM

### *A. Project Rationale*

The GUIDON/NEOMYCIN Project is a research program devoted to the development of a knowledge-based tutoring system for application to medicine. The key issue for the GUIDON/NEOMYCIN project is to develop a program that can provide advice similar in quality to that given by human experts, modeling how they structure their knowledge as well as their problem-solving procedures. The consultation program using this knowledge is called NEOMYCIN. NEOMYCIN's knowledge base, designed for use in a teaching application, is the subject material used by a family of instructional programs referred to collectively as GUIDON2. The problem-solving procedures are developed by running test cases through NEOMYCIN and comparing them to expert behavior. Also, we use NEOMYCIN as a test bed for the explanation capabilities incorporated in our instructional programs.

The purpose of the current contracts is to construct a knowledge-based tutoring system that teaches diagnostic strategies explicitly. By strategy, we mean plans for establishing a set of possible diagnoses, focusing on and confirming individual diagnoses, gathering data, and processing new data. The tutorial program has capabilities to recognize these plans, as well as to articulate strategies in explanations about how to do diagnosis. The strategies represented in the program, modeling techniques, and explanation techniques are wholly separate from the knowledge base, so that they can be used with many medical (and non-medical) domains. That is, the target program will be able to be tested with other knowledge bases, using system-building tools that we provide.

### *B. Medical Relevance and Collaboration*

There is a growing realization that medical knowledge, originally codified for the purpose of computer-based consultations, may be used in additional ways that are medically relevant. Using the knowledge to teach medical students is perhaps foremost among these, and GUIDON2 focuses on methods for augmenting clinical knowledge in order to facilitate its use in a tutorial setting. A particularly important aspect of this work is the insight that has been gained regarding the need to structure knowledge differently, and in more detail, when it is being used for different purposes (e.g., teaching as opposed to clinical decision making). It was this aspect of the GUIDON research that led to the development of NEOMYCIN, which is an evolving computational model of medical diagnostic reasoning that we hope will enable us to better understand and teach diagnosis to students. An important additional realization is that these structuring methods are beneficial for improving the problem-solving performance of consultation programs, providing more detailed and abstract explanations to consultation users, and making knowledge bases easier to maintain.

As we move from technological development of explanation and student modeling capabilities, we are now collaborating closely with medical students and physicians to design an effective, useful tutoring program. In particular, medical students have served as research assistants, and a recent MSAI student is an experienced physician, John Sotos, from Johns Hopkins. The project has also collaborated with a community of researchers focusing on medical education, funded by the Josiah Macy, Jr. Foundation.

## *C. Highlights of Research Progress*

### C.1 Accomplishments This Past Year

### C.1.1 The GUIDON-MANAGE Tutoring Program

This program teaches a student the language of diagnosis by having him or her enter all requests for patient information as an *abstraction*. Thus, the student issues "strategic commands" such as "test the hypothesis meningitis" or "ask a follow-up question about the headache," and the program (NEOMYCIN) carries out the tactics. By year end, this program was operational, with a complex interpreter for simulating NEOMYCIN to generate help, a feedback window to indicate what NEOMYCIN did when it carried out the commands, and many menus for making input to the program convenient. Research continues to focus on the assistance and feedback components of the program.

### C.1.2 Explanation Research

Research in explanation is another major area. This year we adopted a new approach of printing the least possible information that would convey a line of reasoning, rather than generating text to describe everything the program did. The new explanation is based on the idea that the questioner seeking an explanation finds explanations more acceptable if it is necessary to fill in some gaps by his or her own effort. Our current approach is to print the domain relations (e.g., causes or subtype) that link foci to each other, leaving it to the questioner to infer the strategy behind each focus shift. This methodology will allow us to build up a more principled theory of explanation by determining what minimal information is acceptable and what causes an explanation to be inadequate.

### C.1.3 The ODYSSEUS Modeling Program

Our third tutorial-related project involves continued development of a modeling program, ODYSSEUS. The purpose of ODYSSEUS is to discover domain knowledge discrepancies between an application domain knowledge base (e.g. the NEOMYCIN medical knowledge base) and a student or expert problem solver. IMAGE, an earlier modeling program developed in 1982, did not address this problem. The input to ODYSSEUS is the problem solver's patient data requests. When ODYSSEUS watches a student it functions as a student modeling program for GUIDON2 and when it watches an expert it functions as a knowledge acquisition program for HERACLES.

During the past year, a dissertation describing the program has been completed.

### C.1.4 The HERACLES Expert System Shell

The final major effort involves generalizing our expert system tool, HERACLES, so that it can be made available to other research groups who wish to develop knowledge bases which can be tutored by GUIDON2. Several copies of HERACLES were shipped on floppies in the past year to users of Xerox D-Machines.

### C.1.5 Dissemination of Results

Besides publications, a number of tutorials and invited talks by Dr. Clancey presented this work around the world:

- Tutorial Speaker, Evaluation of Expert Systems, AAAI-87, July 1987.
- Tutorial Speaker, Second Advanced Course on AI, Norway, August 1987, six hours of lectures.
- Tutorial Speaker, Knowledge-Based Tutoring, IJCAI-87, Milan, August 1987.
- Invited Speaker, AI-87, Osaka, Japan, October 1987.
- Main Tutorial Speaker, "A Perspective on Knowledge Engineering," Inter Access, The Hague, Netherlands, February 1988, eight hours of lectures.
- "Intelligent Tutoring Systems," panel of the Cognitive Science Society, Seattle, July 1987.
- National Space Sciences Educational Foundation, Stanford, July 1987.
- Fujitsu, Tokyo Japan, CS Forum, November 1987.
- CSK/CRI, Tokyo Japan, CS Forum, November 1987.

## C.2 Research in Progress

As of March, 1988, Dr. William Clancey has moved to the Institute for Research on Learning in Palo Alto, California, where he continues his research on teaching and learning. His use of the SUMEX resource is reduced to accessing archived files.

### *D. Publications Since January 1987*

1. Clancey, W.J. Knowledge-Based Tutoring: The GUIDON Program, Cambridge: MIT Press.
2. Clancey, W.J. From Guidon to Neomycin and Heracles in twenty short lessons: ONR final report, 1979-1985. Current Issues in Expert Systems, 79-123, Academic Press, Inc., London.
3. Clancey, W.J. Intelligent tutoring systems: A tutorial survey. Current Issues in Expert Systems, 39-78, Academic Press, Inc., London.
4. Clancey, W.J. The knowledge engineer as student: Metacognitive bases for asking good questions. In Learning Issues in Intelligent Tutoring Systems, eds. H. Mandl and A. Lesgold, Springer-Verlag, in press. Also KSL 87-12.
5. Wilkins, D.C., Clancey, W.J., and Buchanan, B.G., Knowledge Base Refinement Using Abstract Control Knowledge. January, KSL-87-01.
6. Wilkins, D.C., Buchanan, B.G., and Clancey, W.J., *The Global Credit Assignment Problem and Apprenticeship Learning.* January, KSL-87-04.

*E. Funding Support*

Contract Title: "A Family of Intelligent Tutoring Programs for Medical Diagnosis"
Principal Investigator: Bruce G. Buchanan, Prof. Computer Science, Research
Associate Investigator: William J. Clancey, Research Assoc. Computer Science
Agency: Josiah Macy, Jr. Foundation
Term: March 1985 to March 1988
Total award: $503,415 direct costs

Contract Title: "Computer-Based Tutors for Explaining and Managing the Process of Diagnostic Reasoning"
Principal Investigator: Bruce G. Buchanan, Prof. Computer Science, Research
Associate Investigator: William J. Clancey, Research Assoc. Computer Science
Agency: Office of Naval Research
ID number: N00014-85-K-0305
Term: March 1985 to November 1989
Total award: $712,411 total

## II. INTERACTIONS WITH THE SUMEX-AIM RESOURCE

*A. Medical Collaborations and Program Dissemination via SUMEX*

We are frequently asked to demonstrate GUIDON-MANAGE, GUIDON-WATCH, and NEOMYCIN to Stanford visitors or at meetings in this country or abroad. Physicians have generally been enthusiastic about the potential of these programs and what they reveal about current approaches to computer-based medical decision making. We use network e-mail through SUMEX to communicate with other researchers worldwide.

*B. Sharing and Interaction with Other SUMEX-AIM Projects*

We interact periodically with Paul Feltovich at Southern Illinois Medical School. In addition, the central SUMEX development group acts as an important clearing house for solving problems and distributing new methods.

*C. Critique of Resource Management*

The SUMEX resources group has provided exemplary service. We have no complaints or suggestions whatsoever.

## III. RESEARCH PLANS

*A. Project Goals and Plans*

This research project has now moved from Stanford University to the new Institute for Research on Learning (IRL). We will no longer be an active member of the SUMEX Resource.

*B. Requirements for Continued SUMEX Use*

We have arranged to have archival access to our code and research notes (via dump tapes), which we prepared and stored at SUMEX from 1974 through 1987. We hope to move our personal archived files to our own disks in the next year, but will benefit from continuing access for project files over the next few years.

# IV.A.3. MOLGEN Project

## MOLGEN - Applications of Artificial Intelligence to Molecular Biology: Research in Theory Formation, Testing, and Modification

Prof. E. Feigenbaum
Department of Computer Science
Stanford University

Dr. P. Friedland
NASA-Ames Research Center
Moffett Field, CA

Prof. Charles Yanofsky
Department of Biology
Stanford University

## I. SUMMARY OF RESEARCH PROGRAM

### *A. Project Rationale*

The MOLGEN project has focused on research into the applications of symbolic computation and inference to the field of molecular biology. This has taken the specific form of systems which provide assistance to the experimental scientist in various tasks, the most important of which have been the design of complex experiment plans and the analysis of nucleic acid sequences. Our current research concentrates on scientific discovery within the subdomain of regulatory genetics. We desire to explore the methodologies scientists use to modify, extend, and test theories of genetic regulation, and then emulate that process within a computational system.

Theory or model formation is a fundamental part of scientific research. Scientists both use and form such models dynamically. They are used to predict results (and therefore to suggest experiments to test the model) and also to explain experimental results. Models are extended and revised both as a result of logical conclusions from existing premises and as a result of new experimental evidence.

Theory formation is a difficult cognitive task, and one in which there is substantial scope for intelligent computational assistance. Our research is toward building a system which can form theories to explain experimental evidence, can interact with a scientist to help to suggest experiments to discriminate among competing hypotheses, and can then revise and extend the growing model based upon the results of the experiments.

The MOLGEN project has continuing computer science goals of exploring issues of knowledge representation, problem-solving, discovery, and planning within a real and complex domain. The project operates in a framework of collaboration between the Heuristic Programming Project (HPP) in the Computer Science Department and various domain experts in the departments of Biochemistry, Medicine, and Biology. It draws from the experience of several other projects in the HPP which deal with applications of artificial intelligence to medicine, organic chemistry, and engineering.

### *B. Medical Relevance and Collaboration*

The field of molecular biology is nearing the point where the results of current research will have immediate and important application to the pharmaceutical and chemical industries. Already, clinical testing has begun with synthetic interferon and human growth hormone produced by recombinant DNA technology. Governmental reports estimate that there are more than two hundred new and established industrial firms already undertaking product development using these new genetic tools.

The programs being developed in the MOLGEN project have already proven useful and important to a considerable number of molecular biologists. Currently several dozen researchers in various laboratories at Stanford (Prof. Paul Berg's, Prof. Stanley Cohen's, Prof. Laurence Kedes', Prof. Douglas Brutlag's, Prof. Henry Kaplan's, and Prof. Douglas Wallace's) and over four hundred others throughout the country have used MOLGEN programs over the SUMEX-AIM facility. We have exported some of our programs to users outside the range of our computer network (University of Geneva [Switzerland], Imperial Cancer Research Fund [England], and European Molecular Biology Institute [Heidelberg] are examples). The pioneering work on SUMEX has led to the establishment of a separate NIH-supported facility, BIONET, to serve the academic molecular biology research community with MOLGEN-like software. BIONET is now serving many of the computational needs of over two thousand academic molecular biologists in the United States.

More generally, our work in qualitative simulation as applied to molecular biology is also relevant to building models of many other medical and biological systems. For example, one Artificial Intelligence researcher (Kuipers) has been applying these techniques to the domain of renal physiology. Other researchers within the KSL are using similar techniques to build models of cardio-pulmonary physiology.

### *C. Highlights of Research Progress*

#### *C.1 Accomplishments*

During the past year we have constructed a second model of the tryptophan operon. The first model we built focused on developing qualitative descriptions of the state variables of the tryptophan operon; this second model focuses on the objects in this biological system, their internal structures, and the processes which modify these objects over time. In addition, we have begun to use this second model as the basis for scientific theory formation. The highlights of this work are summarized below.

#### *C.1.1 Qualitative Modeling and Simulation*

Our work in qualitative simulation has been directed towards building a program which embodies a theory of the tryptophan system. The earlier model which we constructed of the system was successful in its ability to predict state variable values for the trp system. However, this system lacked flexibility: its fixed network of state variables is valid for only a limited set of experiments. Many experiments involve introducing new state variables, removing old ones, or modifying the interactions between variables. Thus our goal was to build a model of the system which would essentially derive the state variable network used by the earlier system, given a description of what objects were present in an experiment.

In the newer model a gene regulation experiment is described by specifying what objects are present at the start of the experiment and what their properties and relationships are. These objects are represented as instances of prototypical biological objects which are described in a large knowledge base. The modeling system uses a knowledge base of biological processes to detect interactions between the objects which exist in the current simulation. These interactions can result in the creation of new objects and the establishment of linkages between object state variables, and are used to predict future states of the gene regulation system.

The object knowledge base (KB) is a taxonomic hierarchy of biological objects such as genes, proteins, and chemical binding sites. This KB describes object properties, states, and their decomposition into component objects. The object KB can be viewed as a library of prototypical objects. Modeling of a specific biological experiment begins with a specification of what actual objects (as opposed to prototypes) are present in the experiment. In this context we developed techniques for representing the decomposition of complex objects such as proteins into their components, and for instantiating these prototypical descriptions.

The process knowledge base describes the behaviors of the objects in the trp system. For example, processes encode chemical binding, re-arrangement, and dissociation events which are involved in such biological processes as transcription and biochemical pathways.

When the modeling system is called upon to predict the outcome of an experiment, a process interpreter is responsible for employing processes in the process library to detect interactions between objects in the current experiment. As noted above, these interactions lead to effects such as the creation of new objects in the current experiment, the modification of old objects, and the establishment of linkages between object state variables such as object concentrations.

This model of the trp system has been fully implemented as a working computer program and includes approximately 200 objects and 35 processes. It covers the important components of the trp operon as known in the early 1960s including transcription, translation, and the biosynthetic pathway for tryptophan, and can thus serve as a starting point for the generation of improved theories of the trp operon.

### *C.1.2 Theory Formation*

We have come to view the overall theory formation problem within a machine learning paradigm. Theory formation is considered to be a machine learning problem, which implies that any theory formation program should have two components: a performance element and a learning element. The performance element is the model of the trp operon described above. It contains knowledge of the objects and processes within the trp system, and inference mechanisms for using that knowledge base to predict experimental outcomes. The learning element is used when the performance element makes an incorrect prediction. The learning element must modify the performance element to increase the quality of its predictions.

It appears to be productive to view the theory formation problem as a *design* problem. This allows us to apply knowledge of design within Artificial Intelligence to the problem of theory formation, which is less well understood. Design is a creative activity in which a designer constructs an entity which satisfies a set of constraints. This entity might be an object (e.g., a circuit), or a plan of action (e.g., a robot path plan). The design constraints specify predicates which the design process and the designed entity must satisfy. The entity is constructed from a set of primitives. *Design operators* specify all possible ways in which new primitives may be added to an entity under design. For example, circuits are constructed from transistors and other electronic devices; the design operators describe how these components may be wired together.

AI has approached the problem of design through its central paradigm of search. That is, AI considers the design process to be a kind of search. Search problems have two important aspects: the space to be searched, and the means by which the search is accomplished. In design problems, the *design space* to be searched is the set of all legal configurations in which the design primitives may be combined to produce designed entities.

The task of theory formation also involves the construction of an entity which satisfies a set of constraints. The entity to be designed is a theory. The primitives from which a theory is designed are the objects and processes within the domain. The constraints on the designed theory include: (a) it must account for, or predict, the observed phenomena, (b) it must be testable, (c) it must be confirmed by the tests, (d) it must conform to a large degree with other established knowledge, (e) it must satisfy certain constraints of form, e.g., it should be as simple as possible.

Our earlier historical study of the discovery of attenuation directed us towards this view of theory formation, and it also provided the design operators needed to make it work. In addition to reconstructing the different theories of the trp operon which the biologists possessed at different times, we also compared consecutive theories to determine the differences between them, which tells us what modifications the biologists applied to existing theories to produce new theories. These theory modification operators are used to design new theories from old. Examples of these operators include postulating the presence of a previously known type of object within an experiment, postulating the presence of a previously *unknown* type of object, and postulating an interaction between existing objects which were not previously thought to interact (a new process).

We have begun implementation of a theory formation program based on the above approach. The inputs to this program are (a) the current theory of the trp operon, (b) a description of an experiment, (c) the outcome of the experiment predicted by the theory, and (d) a description of how the prediction in (c) is incorrect. The output of the program is a set of possible modifications to the theory which cause it to make the correct prediction rather than the prediction in (c).

For example, the initial theory might not predict that a certain chemical X is produced in a given experiment, when in fact this chemical is empirically observed to be present. The theory formation program is implemented as an AI planner; in the above example the planner is given the goal of predicting the presence of X. To achieve this goal it can use a number of different theory formation operators, such as postulating the existence of other chemicals in the experiment, and modifying the processes in the theory such that they would cause X to be produced from the chemicals present in the experiment. The current implementation consists of an agenda-based planner with an incomplete set of these theory formation operators.

### *D. Publications*

1. Bach, R., Friedland, P., Brutlag, D., and Kedes, L.: *MAXIMIZE, a DNA sequencing strategy advisor.* Nucleic Acids Res. 10(1):295-304, January, 1982

2. Bach, R., Friedland, P., and Iwasaki, Y.: *Intelligent computational assistance for experiment design.* Nucleic Acids Res. 12(1):11-29, January, 1984.

3. Brutlag, D., Clayton, J., Friedland, P. and Kedes, L.: *SEQ: A nucleotide sequence analysis and recombination system.* Nucleic Acids Res. 10(1):279-294, January, 1982.

4. Clayton, J. and Kedes, L.: *GEL, a DNA sequencing project management system.* Nucleic Acids Res. 10(1):305-321, January, 1982.

Feitelson, J. and Stefik, M.J.: *A case study of the reasoning in a genetics experiment.* Heuristic Programming Project Report HPP-77-18 (working paper), May, 1977.

5. Friedland, P.: *Knowledge-based experiment design in molecular genetics.* Proc. Sixth IJCAI, August, 1979, pp. 285-287.

6. Friedland P.: *Knowledge-based experiment design in molecular genetics.* Stanford Computer Science Report STAN-CS-79-760 (Ph.D. thesis), December, 1979.

7. Friedland, P., Kedes, L. and Brutlag D.: *MOLGEN--Applications of symbolic computation and artificial intelligence to molecular biology.* Proc. Battelle Conference on Genetic Engineering, April, 1981.

8. Friedland, P.: *Acquisition of procedural knowledge from domain experts.* Proc. Seventh IJCAI, August, 1981, pp. 856-861.

9. Friedland, P., Kedes, L., Brutlag, D., Iwasaki, Y. and Bach R.: *GENESIS, a knowledge-based genetic engineering simulation system for representation of genetic data and experiment planning.* Nucleic Acids Res. 10(1):323-340, January, 1982.

10. Friedland, P., and Kedes, L.: *Discovering the secrets of DNA.* Communications of the ACM, 28(11):1164-1186, November, 1985, and IEEE/Computer, 18(11):49:69, November, 1985.

11. Friedland, P. and Iwasaki Y.: *The concept and implementation of skeletal plans.* Journal of Automated Reasoning, 1(2): 161-208, 1985.

12. Friedland, P., Armstrong, P., and Kehler, T.: *The role of computers in biotechnology.* BIO\TECHNOLOGY 565-575, September, 1983.

13. Iwasaki, Y. and Friedland, P.: *SPEX: A second-generation experiment design system.* Proc. of Second National Conference on Artificial Intelligence, August, 1982, pp. 341-344.

14. Martin, N., Friedland, P., King, J. and Stefik, M.J.: *Knowledge base management for experiment planning in molecular genetics.* Proc. Fifth IJCAI, August, 1977, pp. 882-887.

15. Meyers, S. and Friedland, P.: *Knowledge-based simulation of regulatory genetics in bacteriophage Lambda.* Nucleic Acids Res. 12(1):1-9, January, 1984.

16. Stefik, M. and Friedland, P.: *Machine inference for molecular genetics: Methods and applications.* Proc. of NCC, June, 1978.

17. Stefik, M.J. and Martin N.: *A review of knowledge based problem solving as a basis for a genetics experiment designing system.* Stanford Computer Science Report STAN-CS-77-596, March, 1977.

18. Stefik, M.: *Inferring DNA structures from segmentation data: A case study.* Artificial Intelligence 11:85-114, December, 1977.

19. Stefik, M.: *An examination of a frame-structured representation system.* Proc. Sixth IJCAI, August, 1979, pp. 844-852.

20. Stefik, M.: *Planning with constraints.* Stanford Computer Science Report STAN-CS-80-784 (Ph.D. thesis), March, 1980.

21. Karp, P., and D. Wilkins: *An Analysis of the Deep/Shallow Distinction for Expert Systems.* To be published in International Journal of Expert Systems, 1988.

22. Karp, P., and P. Friedland: *Coordinating the Use of Qualitative and Quantitative Knowledge in Declarative Device Modeling*, in Artificial Intelligence, Simulation, and Modeling edited by L. Widman and K. Loparo, 1988.

23. Karp, P.: *A Process-Oriented Model of Bacterial Gene Regulation*, Stanford University Knowledge Systems Laboratory Technical Report KSL-88-18.

24. Round, A.: *QSOPS: A Workbench Environment for the Qualitative Simulation of Physical Processes.* Stanford University Knowledge Systems Laboratory Report KSL-87-37, 1987.

25. Karp, P., and P. Friedland, *A Conceptual Reconstruction of the Discovery of Attenuation.* In Preparation.

### *E. Funding Support*

The MOLGEN grant, which has supported the bulk of this research, is titled: MOLGEN: Applications of Artificial Intelligence to Molecular Biology: Research in Theory Formation, Testing, and Modification. This NSF Grant number MCS-8310236, expired on 10/31/86. The Principal Investigators were Edward A. Feigenbaum, Professor of Computer Science and Charles Yanofsky, Professor of Biology. Additional support for this research has been provided by the Defense Advanced Research Projects Agency, under contract N00039-86C-0033.

## II. INTERACTIONS WITH THE SUMEX-AIM RESOURCE

SUMEX-AIM continues to serve as the nucleus of our computing resources. The facility has not only provided excellent support for our programming efforts but has served as a major communication link among members of the project. Systems available on SUMEX-AIM such as EMACS, MM, Scribe and BULLETIN BOARD have made possible the project's documentation and communication efforts. The interactive environment of the facility is especially important in this type of project development.

We strongly approve of the network-oriented approach to a programming environment into which SUMEX has evolved. The ability to utilize Lisp workstations for intensive computing while still communicate with all of the other SUMEX resources has been indispensable to our work. We currently have a satisfactory mode of operation where essentially all programming takes place on the workstations and most electronic communications, information sharing, and document preparation takes place within the TOPS-20 environment. The evolution of SUMEX has alleviated most of our previous problems with resource loading and file space. Our current Lisp workstations are not quite fast enough, but we are encouraged by the progress that has been made.

We have taken advantage of the collective expertise on medically-oriented knowledge-based systems of the other SUMEX-AIM projects. In addition to especially close ties with other projects at Stanford, we have greatly benefited by interaction with other projects at yearly meetings and through exchange of working papers and ideas over the system.

## III. RESEARCH PLANS

### *A. Project Goals And Plans*

Our current work has the following major goals:

1. The abilities of the theory formation system will be extended by implementing additional theory formation operators to allow it to generate new classes of theories.

2. A mechanism for evaluating alternative theories will be constructed. This mechanism will be guide the planner's search towards more plausible theories, and will allow the system to present only the most credible theories it finds to the user.

3. Test the entire approach on the evolving theory of the trp operon regulatory system. Experiment with different initial knowledge bases to see how the discovery process is altered by the availability of new techniques, analogous systems, and so forth.

### *B. Justification and Requirements for Continued SUMEX Use*

The MOLGEN project depends heavily on the SUMEX facility. We have already developed several useful tools on the facility and are continuing research toward applying the methods of artificial intelligence to the field of molecular biology. The community of potential users is growing nearly exponentially as researchers from most of the biomedical-medical fields become interested in the technology of recombinant DNA. We believe the MOLGEN work is already important to this growing community and will continue to be important. The evidence for this is an already large list of pilot exo-MOLGEN users on SUMEX.

# IV.A.4. ONCOCIN Project

**ONCOCIN Project**

**Edward H. Shortliffe, M.D., Ph.D.**
**Departments of Medicine and Computer Science**
**Stanford University**

**Project Director: Lawrence M. Fagan, M.D., Ph.D**
**Department of Medicine**
**Stanford University**

## I. SUMMARY OF RESEARCH PROGRAM

### *A. Project Rationale*

The ONCOCIN Project is one of many Stanford research programs devoted to the development of knowledge-based expert systems for application to medicine and the allied sciences. The central issue in this work has been to develop a program that can provide advice similar in quality to that given by human experts, and to ensure that the system is easy to use and acceptable to physicians. The work seeks to improve the interactive process, both for the developer of a knowledge-based system, and for the intended end user. In addition, we have emphasized clinical implementation of the developing tool so that we can ascertain the effectiveness of the program's interactive capabilities when it is used by physicians who are caring for patients and are uninvolved in the computer-based research activity.

### *B. Medical Relevance and Collaboration*

The lessons learned in building prior production rule systems have allowed us to create a large oncology protocol management system much more rapidly than was the case when we started to build MYCIN. We introduced ONCOCIN for use by Stanford oncologists in May 1981. This would not have been possible without the active collaboration of Stanford oncologists who helped with the construction of the knowledge base and also kept project computer scientists aware of the psychological and logistical issues related to the operation of a busy outpatient clinic.

### *C. Highlights of Research Progress*

### *C.1.A Background and Overview of Accomplishments*

The ONCOCIN Project is a large interdisciplinary effort that has involved over 35 individuals since the project's inception in July 1979. The work is currently in its ninth year; we summarize here the milestones that have occurred in the research to date:

- *Year 1:* The project began with two programmers (Carli Scott and Miriam Bischoff), a Clinical Specialist (Dr. Bruce Campbell) and students under the direction of Dr. Shortliffe and Dr. Charlotte Jacobs from the Division of Oncology. During the first year of this research (1979-1980), we developed a prototype of the ONCOCIN consultation system, drawing from programs and capabilities developed for the EMYCIN system-building project. During that year, we also undertook a detailed analysis of the day-to-day activities of the Stanford Oncology Clinic in order to determine how to introduce ONCOCIN with minimal disruption of an operation which is already running smoothly. We also spent much of our time in the first year giving careful consideration to the most appropriate mode of interaction with physicians in order to optimize the chances for ONCOCIN to become a useful and accepted tool in this specialized clinical environment.

- *Year 2:* The following year (1980-1981) we completed the development of a special interface program that responds to commands from a customized keypad. We also encoded the rules for one more chemotherapy protocol (oat cell carcinoma of the lung) and updated the Hodgkin's disease protocols when new versions of the documents were released late in 1980; these exercises demonstrated the generality and flexibility of the representation scheme we had devised. Software protocols were developed for achieving communication between the interface program and the reasoning program, and we coordinated the printing routines needed to produce hard copy flow sheets, patient summaries, and encounter sheets. Finally, lines were installed in the Stanford Oncology Day Care Center, and, beginning in May 1981, eight fellows in oncology began using the system three mornings per week for management of their patients enrolled in lymphoma chemotherapy protocols.

- *Year 3:* During our third year (1981-1982) the results of our early experience with physician users guided both our basic and applied work. We designed and began to collect data for three formal studies to evaluate the impact of ONCOCIN in the clinic. This latter task required special software development to generate special flow sheets and to maintain the records needed for the data analysis. Towards the end of 1982 we also began new research into a *critiquing model* for ONCOCIN that involves "hypothesis assessment" rather than formal advice giving. Finally, in 1982 we began to develop a query system to allow system builders as well as end users to examine the growing complex knowledge base of the program.

- *Year 4:* Our fourth year (1982-1983) saw the departure of Carli Scott, a key figure in the initial design and implementation of ONCOCIN, the promotion of Miriam Bischoff to Chief Programmer, and the arrival of Christopher Lane as our second scientific programmer. At this time we began exploring the possibility of running ONCOCIN on a single-user professional workstation and experimented with different options for data-entry using a "mouse" pointing device. Christopher Lane became an expert on the Xerox workstations that we are using. In addition, since ONCOCIN had grown to such a large program with many different facets, we spent much of our fourth year documenting the system. During that year we also modified the clinic system based upon feedback from the physician-users, made some modifications to the rules for Hodgkin's disease based upon changes to the protocols, and completed several evaluation studies.

- *Year 5:* The project's fifth year (1983-1984) was characterized by growth in the size of our staff (three new full-time staff members and a new oncologist joined the group). The increased size resulted from a DRR grant that permitted us to begin a major effort to rewrite ONCOCIN to run on professional workstations. Dr. Robert Carlson, who had been our Clinical Specialist for the previous two years, was replaced by Dr. Joel Bernstein, while Dr. Carlson assumed a position with the nearby Northern California Oncology Group; this appointment permitted him to continue his affiliation both with Stanford and with our research group. In August of 1983, Larry Fagan joined the project to take over the duties of the ONCOCIN Project Director while also becoming the Co-Director of the newly formed Medical Information Sciences Program. Dr. Fagan continues to be in charge of the day-to-day efforts of our research. An additional programmer, Jay Ferguson, joined the group in the fall to assist with the effort required to transfer ONCOCIN from SUMEX to the 1108 workstation. A fourth programmer, Joan Differding, joined the staff to work on our protocol acquisition effort (OPAL).

- *Year 6:* During our sixth year (1984-1985) we further increased the size of our programming staff to help in the major workstation conversion effort. The ONCOCIN and OPAL efforts were greatly facilitated by a successful application for an equipment grant from Xerox Corporation. With a total of 15 Xerox LISP machines now available for our group's research, all full-time programmers have dedicated machines, as do several of the senior graduate students working on the project. Christopher Lane took on full-time responsibility for the integration and maintenance of the group's equipment and associated software. Two of our programming staff moved on to jobs in industry (Bischoff and Ferguson) and three new programmers (David Combs, Cliff Wulfman, and Samson Tu) were hired to fill the void created by their departure and by the reassignment of Christopher Lane.

  In addition to funding from DRR for the workstation conversion effort, we have support from the National Library of Medicine which supports our more basic research activities regarding biomedical knowledge representation, knowledge acquisition, therapy planning, and explanation as it relates to the ONCOCIN task domain. We have continued to study the therapy planning process under support from the NLM. This research is led by Dr. Fagan and has concentrated on how to represent the therapy-planning strategies used to decide treatment for patients who run into serious problems while on protocol-described treatment. The physicians who treat these patients often seek out a consultation with the protocol study chairman. Dr. Branimir Sikic, a faculty member from the Stanford University Department of Medicine, and the Study Chairman for the oat cell protocol, collaborated on this project. Janice Rohn joined the ONCOCIN project as data manager and to assist in the knowledge entry process.

- *Year 7:* The seventh year (1985-86) marked several milestones in our research on workstation-based programming. The OPAL knowledge acquisition system became operational, and several new oncology protocols were entered using this system. David Combs was primarily responsible for creating the operational version of OPAL (based on the initial prototype by Joan Differding Walton). As anticipated, we increased the speed and ease with which protocols can be added to the ONCOCIN knowledge base.

  Based on the protocols entered through OPAL, we began experimental testing of the workstation version of ONCOCIN in the Stanford oncology clinic. Clifford Wulfman developed the user interface (based on an initial prototype designed by Christopher Lane). Samson Tu developed the reasoning component (designed originally by Jay Ferguson). Much of their work is built upon an object-oriented system developed for our group by Christopher Lane. We connected the various parts of the system, and demonstrated that we have the capability to run ONCOCIN with the reasoning program and interface program on different machines in the communication network. The current version of the program is currently run on a single workstation, but future versions may take advantage of the multiple machine option. To increase the speed at which we are able to test protocols entered into ONCOCIN, we developed additional programs to test real and synthetic cases without user interaction; these are then reviewed by our collaborating clinicians.

We also developed a workstation-based program, OPUS, to help clinicians determine which protocols are appropriate for specific patients. OPUS was designed and implemented by Janice Rohn with the assistance of Christopher Lane. We have been using it in the clinic setting since the end of 1985. Thus, in addition to providing an information resource about protocols, the use of a graphically-oriented program provided a way to learn about the software style and hardware used in the workstation version of ONCOCIN.

We discontinued the mainframe version of ONCOCIN, and began using the workstation version exclusively. The performance of the mainframe version of ONCOCIN was documented in two evaluation papers that appeared in clinical journals (see Hickam and Kent's papers).

We continued our basic research in the design of advanced therapy-planning programs: the ONYX project. We developed a model for planning which includes techniques from the fields of artificial intelligence, simulation, and decision analysis. Artificial intelligence techniques are used to create a small number of possible plans given the ideal therapy and the patient's past treatment history. Simulation techniques and decision analysis are used to examine and order the most promising plans. Our goal is to allow ONCOCIN to give advice in a wider range of situations; in particular, the system should be able to recommend plans for patients who have an unusual response to chemotherapy.

During this year, Stephen Rappaport, M.D. joined us as a programmer on the therapy planning research. Clinical expertise for ONCOCIN was provided by Richard Lenon, M.D. and Robert Carlson, M.D.

- *Year 8:* The eight year (1986-87) concentrated on two diverse tasks: 1) scaling up the use of the workstation version of ONCOCIN in the clinic, and 2) generalization of each of the components. The latter task is described in the core research sections of this report(see page 19).

In 1986, we placed the workstation version of ONCOCIN into the Oncology Day Care clinic. This version is a completely different program from the version of ONCOCIN that ran on the DECsystem 20--using protocols entered through the OPAL program, with a new graphical data entry interface, and a revised knowledge representation and reasoning component. One of the Oncology Clinical Fellows (Andy Zelenetz) became responsible for verifying how well our design goals for ONCOCIN had been accomplished. His suggestions have included the addition of key protocols and the ability to have the program used as a data management tool if the complete treatment protocol had not yet been entered into the system. Both of these suggestions were carried out during this year, and the program has achieved wider use in the clinic setting. In addition, laser-printed flowsheets and progress notes have been added to the clinic system.

The process of entering a large number of treatment protocols in a short period of time led to other research topics including: design of an automated system for producing meaningful test cases for each knowledge base, modification of the design and access methods for the time-oriented database, and the development of methods for graphically viewing multiple protocols that are combined into one large knowledge base. These research efforts will continue into the next year. In addition, some of the treatment regimens developed for the original mainframe version are still in use and can be transferred to the [illegible] version of ONCOCIN. As the knowledge base grows, additional mechanisms will be needed for the incremental update and retraction of protocols. Additional changes in the reasoning and interface components of the system are described below.

A new research project related to ONCOCIN was started this last year. We are exploring the use of continuous speech recognition as an alternate entry method for communicating with ONCOCIN. This project requires the connection of speech recognition equipment produced by Speech Systems, Inc. of Tarzana to the ONCOCIN interface module. Christopher Lane has developed a prototype network connection and command interpreter between the speech module (running on a Sun with special hardware added) and the Xerox 1186 computer that runs ONCOCIN. Clifford Wulfman has designed a series of modifications to the ONCOCIN user interface to allow for verbal commands. This work is described in more detail in the core ONCOCIN section.

We continue to collaborate with Andy Zelenetz, Richard Lenon, Robert Carlson, and Charlotte Jacobs on the design and implementation of ONCOCIN in the clinic. Stephen Rappaport has started a residency program to continue his medical education.

- *Year 9:* The majority of our effort this year has been to understand the limitations of the clinic version of ONCOCIN, and to concentrate on the generalization of these techniques to other application areas besides oncology. The majority of this research is thus described as part of the core research discussion on ONCOCIN. Highlights of this year include: (1) development of a general knowledge acquisition tool (PROTEGE) designed to handle skeletal planning applications for clinical trials in any area of medicine, (2) demonstration that the therapy planning and knowledge acquisition tools for ONCOCIN can be closely integrated, and (3) development of a speech input system for ONCOCIN.

As a demonstration of the capabilities of the project to date, we undertook an experiment to see how difficult and time-consuming it is to bring up a new treatment protocol. A summary of a recent colon protocol was downloaded from the PDQ protocol database. Approximately 60% of the knowledge of the protocol summary fit easily into the OPAL high level description. Additional rules were entered using lower level editors. A limited consultation was run after about 4 hours of work. Although this is only one data point, we believe that it validates the generality of the knowledge acquisition and therapy planning approach that we have pursued for nearly a decade. Work continues on extending the knowledge acquisition and therapy planning tools to allow for a higher percentage of concepts that can be entered with the smallest possible amount of low level Lisp changes.

Although we have completed the transfer of ONCOCIN into a stable and useful system on the Xerox Lisp workstations, it is now clear that this type of machine will not provide the type of dissemination hardware we would like to see. There are no planned additions to increase the speed, decrease the cost, or increase the integration capabilities of these workstations. Although there may be other solutions that will allow us to port ONCOCIN directly to alternative hardware platforms, we may need to move away from Xerox workstations and InterLisp language upon which most of our software is based. We are particularly interested in exploring the Mac II hardware recently purchased for the KSL.

*C.1.B Review of Research Issues in ONCOCIN and OPAL*

Our work to refine the clinic versions of ONCOCIN and OPAL reached a mature stage during this last research year. As our attention has moved to the generalization of these tasks (E-ONCOCIN and PROTEGE) it seems appropriate to describe the range of research issues that we have examined during the development of the ONCOCIN system.

*Research Issues in the Development of the ONCOCIN Reasoner and Interviewer*

- *Redesign of the reasoning component.* A major impetus for the redesign of the system was to develop more efficient methods to search the knowledge base during the running of a case. We have implemented a reasoning program that uses a discrimination network to process the cancer protocols. This network provides for a compact representation of information which is common to many protocols but does not require the program to consider and then disregard information related to protocols that are irrelevant to a particular patient. We continue to improve portions of the reasoning component that are associated with reasoning over time; e.g., modeling the appropriate timing for ordering tests and identifying the information which needs to be gathered before the next clinic visit. In general, we are concentrating on improving the representation of the knowledge regarding sequences of therapy actions specified by the protocol.

  Our experience with adding a large number of protocols has led to the evaluation of the design of the internal structure of the knowledge base (e.g., the way we describe the relationships between chemotherapies, drugs, and treatment visits). We will continue to improve the method for traversing the plan structure in the knowledge base, and consider alternative arrangements for representing the structure of chemotherapy plans. Currently, the knowledge base of treatment guidelines and the patient database are separated. We propose to tie these two structures closer together. Additional work is anticipated on turning ONCOCIN into a critiquing system, where the physician enters their therapy and ONCOCIN provides suggestions about possible alternatives to the entered therapy. Although we have concentrated our review of the ONCOCIN design primarily on the data provided by additional protocols, we know that non-cancer therapy problems may also raise similar issues. The E-ONCOCIN effort is designed to produce a domain-independent therapy planning system that includes the lessons learned from our oncology research. Samson Tu is primarily responsible for continued improvement of the reasoning component of ONCOCIN.

- *Development of a temporal network.* The ability to represent temporal information is a key element of programs that must reason about treatment protocols. The earlier version of the ONCOCIN system did not have an explicit structure for reasoning about time-oriented events. We are experimenting with different configurations of the temporal network, and with the syntax for querying the network. We are also adapting this network so that it can interface with the ONYX therapy-planning systems. This research on temporal reasoning is part of Michael Kahn's Ph.D. thesis. Michael is a student in the Medical Information Sciences Program at University of California at San Francisco.

- *Extensions to the user interface.* We continue to experiment with various configurations of the user interface. Many of the changes have been in response to requests for a more flexible data management environment. We are occasionally faced with data that becomes available corresponding to a time before the current visit. This can happen if a laboratory result is delayed, or a patient's electronic flowsheet is started in the middle of the treatment. We have added the ability to create new columns of data, and are designing the changes to the temporal processing components of ONCOCIN to allow for data that is inserted out of order. We have also extended the flowsheet to allow for patient specific parameters (e.g., special test results or symptoms) that the physician wishes to follow over time. The flowsheet layouts have been modified to create protocol specific flowsheets, e.g., lymphoma flowsheets have a different configuration than lung cancer flowsheets. The basic structure of the interface has been modified to use object-oriented methods, which allows for more flexible interaction between different components of the flowsheet and the operations performed on the flowsheet.

A continuing area of research concerns how to guide the user to the most appropriate items to enter (based on the needs of the reasoning program) without disrupting the fixed layout of the flowsheet. The mainframe version of ONCOCIN modified the order of items on the flowsheet to extract necessary information from the user. In the workstation version, we have developed a guidance mechanism which alerts the user to items that are needed by the reasoning program. The user is not required to deviate from a preferred order of entry nor required to respond to a question for which no current answer is available. Cliff Wulfman is primarily responsible for improvements to the user interface of ONCOCIN.

- *System support for the reorganization.* The LISP language, which we used to build the first version of ONCOCIN, does not explicitly support basic knowledge manipulation techniques (such as message passing, inheritance techniques, or other object-oriented programming structures). These facilities are available in some commercial products, but none of the existing commercial implementations provide the reliability, speed, size, or special memory-manipulation techniques that are needed for our project. We have therefore developed a "minimal" object-oriented system to meet our specifications. The object system is currently in use by each component of the new version of ONCOCIN and in the software used to connect these components. In addition, all ONCOCIN student projects are now based on this programming environment. Christopher Lane created and is responsible for modifications to the object-oriented system.

*Interactive Entry of Chemotherapy Protocols by Oncologists (OPAL)*

We continue to refine the software (the OPAL system) that permits physicians who are not computer programmers to enter protocol information on a structured set of forms presented on a graphics display. Most expert systems require tedious entry of the system's knowledge. In many other medical expert systems, each segment of knowledge is transferred from the physician to the programmer, who then enters the knowledge into the expert system. We have taken advantage of the generally well-structured nature of cancer treatment plans to design a knowledge entry program that can be used directly by clinicians. The structure of cancer treatment plans includes:

- choosing among multiple protocols (that may be related to each other);
- describing experimental research arms in each protocol;
- specifying individual drugs and drug combinations;

- setting the drug dosage level;
- and modifying either the choice of drugs or their dosage.

Using the graphics-oriented workstations, this information is presented to the user as computer-generated forms which appear on the screen. After the user fills in the blanks on the forms, the program generates the rules used to drive the reasoning process. As the user describes more detailed aspects of the protocol, new forms are added to the computer display; these allow the user to specify the special cases that make the protocols so complicated. Although the user is unaware of the creation of the knowledge base from the interaction with OPAL, a complex set of translations are taking place. The user's entries are mapped into an intermediate data structure (IDS) that is common for all protocols. From the IDS, a translation program generates rules for creating and modifying treatment, and integrates them with the existing ONCOCIN knowledge base. Considerable effort has been expended on producing a standard relational database as the appropriate data structure to underlie the OPAL IDS. The PROTEGE system described in the core ONCOCIN section was built upon this relational database.

Although the "forms" were specifically designed for cancer treatment plans, the techniques used to organize data can be extended to other clinical trials, and eventually to other structured decision tasks. The key factor is to exploit the regularities in the structure of the task (e.g., this interface has an extensive notion of how chemotherapy regimens are constructed) rather than to try to build a knowledge-entry system that can accept *any* possible problem specification. The OPAL program is based upon a domain-independent forms creation package designed and implemented by David Combs. This program will provide the basis for our extension of OPAL to other application areas.

We have now entered thirty-five protocols covering many different organ systems and styles of protocol design. Based on this experience, we continue to explore ways to modify OPAL to increase the percentage of the protocol that can be entered directly by our clinical collaborators. One direction in which we have extended the OPAL program is in providing a graphical interface of nodes and arcs to specify the procedural knowledge about the order of treatments and important decision points within the treatments. This work is described in several papers by Musen.

### *C.2 Research in Progress*

The major thrusts in speech input and generalized knowledge acquisition are described in the core research description of ONCOCIN. We will describe here our research in complex therapy planning and it's spin-offs in temporal representations and summarization of patient records.

#### *C.2.1 Strategic Therapy Planning (ONYX)*

As mentioned above, we have continued our research project (ONYX) to study the therapy-planning process and to determine how clinical strategies are used to plan therapy in unusual situations. Our goals for ONYX are: (1) to conduct basic research into the possible representations of the therapy-planning process, (2) to develop a computer program to represent this process, and (3) eventually to interface the planning program with ONCOCIN. We have worked with our clinical collaborators to determine how to create therapy plans for patients whose special clinical situation preclude following the standard therapeutic plan described in the protocol document.

The prototype program design has four components: (1) to review the patient's past record and recognize emerging problems, (2) to formulate a small number of revised therapy plans based on existing problems, (3) to determine the results of the generated plans by using simulation, and (4) to weight the results of the simulation and rank order the plans by performing decision analysis. This model is described in the papers by Langlotz.

We have built an expert system based on decision analytic techniques as part of the solution to the fourth step of the ONYX planning problem. The program carries out a dialogue with the user concerning the particular treatment choices to be compared, potential problems with the treatments, and the patient-specific utilities corresponding to the possible outcomes. A decision tree is automatically created, displayed on the screen, and solved. The solution is presented to the user, and is compatible with a explanation program for decision trees being developed as part of the Ph.D. research of Curtis Langlotz.

A major spin-off from our ONYX work is a program that can summarize temporal trends in patient visits during chemotherapy and produce a summary of the patient's course using both data from the flowsheet and an *underlying model of bone marrow physiology*. This work has led to major improvements in the temporal representation and in the integrate of mathematical and symbolic models. This work is part of Michael Kahn's Ph.D. thesis.

Summarization is defined as the task of combining multiple observations or features into a more general statement and abstraction as the task of selecting a subset of available features considered most relevant to answering a particular question. Both tasks require a model of the underlying system that encodes extensive knowledge about the entities and relationships that cause the system behavior and result in the observations. In the setting of a dynamic system, the model must be capable of representing temporal relationships between entities.

This work proposes that the combination of mathematical and symbolic techniques can be used to construct useful summaries of complex time-ordered data. In particular, mathematical models are used to capture the knowledge about the physiological processes that are responsible for the patient's clinical findings. Model parameters represent physiological concepts that are clinically relevant for medical problem solving. Prior to any patient-specific observations, the model parameters are set to population-based estimates. Standard curve-fitting techniques using a Bayesian updating scheme adjust model parameters to new observations. As more patient-specific observations are obtained, the set of estimated model parameters move further away from the population estimates. Symbolic models are used to augment the mathematical model parameter and state estimates. As the patient's clinical course evolves, the symbolic model captures the concurrent contexts that affect the interpretation of the physiological model results. For example, a heart rate of 120 is considered abnormally high in the context of a resting person but may be inappropriately low in the context of a treadmill stress test. A key feature of the combined mathematical and symbolic approach is that the physiological model changes over time as additional data are obtained and the symbolic model modifies the interpretation of these model changes in light of the clinical contexts present when the data was observed.

The methodology for combining mathematical and symbolic models emphasizes four main elements in summarizing complex time-ordered data:

1. A mechanistically-motivated model (in medicine, a physiological model) forms the basis for converting raw observations into more meaningful concepts. However, the interpretation of these concepts requires additional knowledge such as the contextual information contained in a symbolic model.

2. The initial model is based on general knowledge since no specific observations are available to alter the initial impression. New observations will change the initial model by incorporating the new information. The collection of altered models captures state changes that have evolved over time.

3. Differences in key model features or states form the basis for selective abstraction and effective summarization. A method for determining which features are pertinent to a user question or sufficiently "interesting" to warrant inclusion into a summarization requires additional domain-specific reasoning.

4. The construction of a concise and useful summarization requires the use of additional contextual and domain-specific information so that the generated summary text conforms to the user's expectations and requirements.

These principles form the basis for a computer program designed to summarize the clinical course of individual patients receiving experimental cancer chemotherapy. In this setting, patients are often receiving more than one treatment that have overlapping schedules and durations of action. Thus our temporal model requires the representation and the reasoning with multiple, simultaneous contexts to ensure the proper interpretation of a given observation or model estimate. ONCOCIN uses a specialized structure called the temporal network to represent treatment contexts used in temporal queries into a time-oriented patient record. We have extended the temporal network concept to create a symbolic model of the patient's clinical course over time. This structure permits the representation of multiple, concurrent contexts over time and therefore can capture the complex temporal nature of our patient's clinical course. For the proper interpretation of the mathematical model output, the temporal network provides the set of contexts that existed when the observation and model estimates were obtained. In addition, the interpretation task requires complex context-sensitive reasoning. For example, the interpretation of a model parameter may be different if two contexts were present concurrently than if either context was present alone. The temporal network provides a mechanism for altering the available reasoning methods based on the set of current contexts. In this use of the temporal network, reasoning methods are associated with each context. When a context is present, a temporal network node representing that context is created and the reasoning methods are made available to the interpretation process. A temporal network node may also withdraw methods made available by other temporal network nodes. In this manner, a general rule or method can be suspended if it is not appropriate in particular context.

We believe that the combination of mathematical models along with specialized symbolic structures results in more representational and inferencing power than either method alone. Well established mathematical techniques convert observations into underlying system concepts while symbolic techniques interpret the mathematical results using additional domain knowledge. Although some of these features could possibly be represented using either mathematical or symbolic techniques alone, we believe the resulting "pure" models would be too complex to be useful. In combining the two approaches, concepts best modeled in a physiological paradigm can be expressed within the mathematical model while concepts best modeled symbolically can be represented within the temporal network.

### *C.2.2 Documentation*

In 1986, we videotaped a lecture and demonstration of the ONCOCIN and OPAL systems at the XEROX Palo Alto Research Center. This videotape is available for loan from our offices. Our previous videotapes have been shown at scientific meetings and have been distributed to many researchers in other countries. The publications described below further document our recent work on ONCOCIN. We have sent copies of our system to the University of Pittsburgh, and will distribute it to the National Library of Medicine. We have developed a user manual, description of sample interaction, reference card, and graphical flowchart to help with training in the use of ONCOCIN.

### *D. Publications Since January, 1987*

1. Walton, J.D., Musen, M.A., Combs, D., Lane, C.D., Shortliffe, E.H., and Fagan, L.M. Graphical access to medical expert systems: III. Design of a knowledge-acquisition environment. Memo KSL-85-30. Methods of Information in Medicine, 26:78-88, 1987.

2. Musen, M.A., Fagan, L.M., and Shortliffe, E.H. Graphical specification of procedural knowledge for an expert system. Memo KSL-85-53. Presented at the Second IEEE Computer Society Workshop on Visual Languages, pp. 167-178, Dallas, TX, June 1986. Reprinted in Expert Systems: The User Interface (J. Hendler, ed.), pp. 15-35 (Chapter 2), Norwood, NJ: Ablex Publishing Company, 1988.

3. Langlotz, C.P., Fagan, L.M., Tu, S.W., Sikic, B.I., and Shortliffe, E.H. A therapy planning architecture that combines decision theory and artificial intelligence techniques. KSL-85-55. Computers in Biomedical Research, 20:279-303, 1987.

4. Musen, M.A., Fagan, L.M., Combs, D.M., and Shortliffe, E.H. Use of a domain model to drive an interactive knowledge-editing tool (Memo KSL-86-24). International Journal of Man-Machine Studies 26(1):105-121, 1987.

5. Shortliffe, E.H. Artificial Intelligence in Management Decisions: ONCOCIN. Memo KSL-86-39. Proceedings of a Conference on Medical Information Sciences, University of Texas Health Sciences Center at San Antonio, July 1985. Also to appear in Frontiers of Medical Information Sciences, Praeger Publishing, 1988.

6. Langlotz, C.P., Shortliffe, E.H., and Fagan, L.M. A methodology for generating computer-based explanations of decision-theoretic advice. Technical report, KSL-86-57, July 1987. To appear in Medical Decision Making.

7. Langlotz, C.P. and Shortliffe, E.H. An analysis of logic and decision-theoretic methods for planning under uncertainty. Report KSL-87-17, 1987.

8. Shortliffe, E.H. Computer programs to support clinical decision making. Memo KSL-87-30. Journal of the American Medical Association, 258:61-67, 1987.

9. Rennels, G.D. and Shortliffe, E.H. Advanced computing for medicine (Memo KSL-87-33). Scientific American, pp. 154-161, October 1987.

10. Langlotz, C.P. Advice generation in an axiomatically-based expert system. Proceedings of the Eleventh Annual Symposium on Computer Applications in Medical Care, pp. 49-55, Washington, D.C. 1987.

11. Shortliffe, E.H. and Hubbard, S.M. Information systems in oncology. Memo KSL-87-66, November 1987. To appear as a chapter in Cancer: Principles and Practice of Oncology (V.T. DeVita, S. Hellman, and S.A. Rosenberg, eds.), 1988.

12. Musen, M.A. Generation of Model-Based Knowledge-Acquisition Tools for Clinical-Trial Advice Systems. KSL-88-06. Doctoral dissertation, Medical Information Sciences Program, Medical Computer Science Group, Stanford University, January 1988.

13. Wulfman, C.E., Isaacs, E.A., Webber, B.L., and Fagan, L.M. Integration discontinuity: Interfacing users and systems. Memo KSL-88-12, February 1988. Proceedings of Architectures for Intelligent Interfaces: Elements and Prototypes, pp. 57-68, Monterey CA, March 29-April 1, 1988.

14. Musen, M.A. Conceptual models of interactive knowledge-acquisition tools. Report KSL-88-16, March 1988.

15. Musen, M.A. Generation of knowledge-acquisition tools from clinical-trial models. Report KSL-88-26, March 1988. To be published in The Proceedings of Medical Informatics, Europe 1988, Oslo, Norway, August 1988.

*E. Funding Support*

Grant Title: "Therapy-planning strategies for consultation by computer"
Principal Investigator: Edward H. Shortliffe
Project Management: Lawrence M. Fagan
Agency: National Library of Medicine
ID Number: LM-04136
Term: April 1987 to March 1990
Total award: $380,123

Grant Title: "Knowledge Management for Clinical Trial Advice Systems"
Principal Investigator: Edward H. Shortliffe
Project Management: Lawrence M. Fagan
Agency: National Library of Medicine
ID Number: 1 R01 LM04420-01
Term: September 1985 through August 1988
Total award: $314,707

Grant Title: Postdoctoral Training in Medical Information Science
Principal Investigator: Edward H. Shortliffe
Project Management: Edward H. Shortliffe
Agency: National Library of Medicine
ID Number: 1 T32 LM07033
Term: July 1, 1984 - June 30, 1989
Total award: $903,718

Grant Title: Henry J. Kaiser Faculty Scholar in General Internal Medicine
Principal Investigator: Edward H. Shortliffe
Agency: Henry J. Kaiser Family Foundation
Term: July 1983 to June 1988
Total award: $250,000 ($50,000 annually).

Grant Title: Explanation of Computer-assisted therapy plans
Principal Investigator: Lawrence M. Fagan
Agency: National Institutes of Health
ID Number: 1 R23 LM04316
Term: 2/1985-1/1988
Total award: $107,441

## II. INTERACTIONS WITH THE SUMEX-AIM RESOURCE

### *A. Medical Collaborations and Program Dissemination via SUMEX*

A great deal of interest in ONCOCIN has been shown by the medical, computer science, and lay communities. We are frequently asked to demonstrate the program to Stanford visitors. We also demonstrated our developing workstation code in the Xerox exhibit in the trade show associated with AAAI-84 in Austin, Texas, IJCAI-85 in Los Angeles, AAAI-86 in Philadelphia, and Medinfo 86. Physicians have generally been enthusiastic about ONCOCIN's potential. The interest of the lay community is reflected in the frequent requests for magazine interviews and television coverage of the work. Articles about MYCIN and ONCOCIN have appeared in such diverse publications as *Time* and *Fortune*, and ONCOCIN has been featured on the "NBC Nightly News," the PBS "Health Notes" series, and "The MacNeil-Lehrer Report." Most recently it appeared in a special on Artificial Intelligence for TV Ontario (Canadian PBS station) and "Physician's Journal Update" on the Lifetime Cable Network. Due to the frequent requests for ONCOCIN demonstrations, we have produced a videotape about the ONCOCIN research which includes demonstrations ONCOCIN and OPAL. The tape has also been shown to both national and international researchers in biomedical computing. We are producing a tape that describes our speech research.

Our group also continues to oversee the MYCIN program (not an active research project since 1978) and the EMYCIN program. Both systems continue to be in demand as demonstrations of expert systems technology. MYCIN has been demonstrated via networks at both national and international meetings in the past, and several medical school and computer science teachers continue to use the program in their computer science or medical computing courses. Researchers who visit our laboratory often begin their introduction by experimenting with the MYCIN/EMYCIN systems. We also have made the MYCIN program available to researchers around the world who access SUMEX using the GUEST account. EMYCIN has been made available to interested researchers developing expert systems who access SUMEX via the CONSULT account. One such consultation system for psychopharmacological treatment of depression, called Blue-Box (developed by two French medical students, Benoit Mulsant and David Servan-Schreiber), was reported in July of 1983 in *Computers and Biomedical Research.* The EMYCIN experience is now well disseminated via commercial products. We may be able to demonstrate MYCIN from the 2020 after the 2060 is turned off, but this is not a major concern in the transition effort.

### *B. Sharing and Interaction with Other SUMEX-AIM Projects*

The community created on the SUMEX resource has other benefits which go beyond actual shared computing. Because we are able to experiment with other developing systems, such as INTERNIST/CADUCEUS, and because we frequently interact with other workers (at AIM Workshops or at other meetings), many of us have found the scientific exchange and stimulation to be heightened. Several of us have visited workers at other sites, sometimes for extended periods, in order to pursue further issues which have arisen through SUMEX- or workshop-based interactions. In this regard, the ability to exchange messages with other workers, both on SUMEX and at other sites, has been crucial to rapid and efficient dissemination of ideas. Certainly it is unusual for a small community of researchers with similar scholarly interests to have at their disposal such powerful and efficient communication mechanisms, even among those researchers on opposite coasts of the country.

During this past three years, we have had extensive interactions with Randy Miller at Pittsburgh. Via floppy disks and SUMEX, we have experimented with several versions of the QMR program. The interaction was very much facilitated by the availability of SUMEX for communication and data transmission. Several recent papers have been written to describe collaborations between students in our training program and the group at the University of Pittsburgh and Carnegie-Mellon University.

### *C. Critique of Resource Management*

Our community of researchers has been extremely fortunate to work on a facility that has continued to maintain the high standards that we have praised in the past. The staff members are always helpful and friendly, and work as diligently to please the SUMEX community as to please themselves. As a result, the computer is as accessible and easy-to-use as they can make it. More importantly, it is a reliable and convenient research tool. We extend special thanks to Tom Rindfleisch for maintaining such high professional standards. As our computing needs grow, we have increased our dependence on special SUMEX skills such as networking and communication protocols. As described above, we will eventually be moving our software development to a combination of Lisp machines and Mac II's. These will need to be easily networked together while still providing the communications resources of the DEC-20.

## III. RESEARCH PLANS

### *A. Project Goals and Plans*

In the coming year, there are several areas in which we expect to expend our efforts on the ONCOCIN System:

1. *To generalize the reasoning and interaction components of the ONCOCIN system for other applications.*

2. *Extensions and generalizations of the OPAL and PROTEGE knowledge acquisition experiments*

3. *Extensions to the strategic planning framework including research on temporal representations, mathematical modeling and summarization*

4. *To continue testing of the workstation version of ONCOCIN.*

### *B. Justification and Requirements for Continued SUMEX Use*

All of our research takes place in the context of the SUMEX resource. The development of our project will continue to take place on LISP machines which we have purchased or which have been donated by the XEROX Corporation, with a gradual transition to Mac II's. The word processing and communication requirements will be met by the Mac II's plus the excellent network services provided by SUMEX.

*C. Requirements for Additional Computing Resources*

Most of our CPU needs are met with our current equipment. However, our requirements for high bandwidth communication facilities are increased as we have an increasingly distributed environment. We would appreciate support that provides equivalent resources to what we are used to on the DEC-20 including mail support, and file servers. We will continue to need access to the international networks that we now use for much of our communication with colleagues. For example, Mark Musen (who recently finished his Ph.D., and will be an Assistant Professor in our department when he returns) is currently in Holland for 9 months. During this period, we have been in constant electronic communication with him over a set of networks. This has been a significant asset to our research project.

*D. Recommendations for Future Community and Resource Development*

The continuation of network support and file service are our major needs. Help with software selection and implementation for our Mac II's is also an important requirement to maintain research continuity as the DEC-20 is disconnected. Maintaining the excellent mail service and access to the international networks is also essential for our research.

# IV.A.5. PROTEAN Project

## PROTEAN Project

Oleg Jardetzky
Nuclear Magnetic Resonance Lab, School of Medicine
Stanford University

Bruce Buchanan, Ph.D.
Computer Science Department
Stanford University

## I. SUMMARY OF RESEARCH PROGRAM

### *A. Project Rationale*

The goals of this project are related both to biochemistry and artificial intelligence: (a) use existing AI methods to aid in the determination of the 3-dimensional structure of proteins in solution, and (b) use protein structure determination as a test problem for experiments with large scale constraint satisfaction, an area of increasing interest to artificial intelligence. Empirical data from nuclear magnetic resonance (NMR) and other sources may provide enough constraints on structural descriptions to allow protein chemists to bypass the laborious methods of crystallizing a protein and using X-ray crystallography to determine its structure. This problem exhibits considerable computational complexity, but by formulating it as search problem, it should be possible to utilize many of the knowledge-based techniques developed in AI for dealing with large search spaces.

### *B. Medical Relevance*

The molecular structure of proteins is essential for understanding many problems of medicine at the molecular level, such as the mechanisms of drug action. Using NMR data from proteins in solution will allow the study of proteins whose structure cannot be determined with other techniques, and will decrease the time needed for the determination.

### *C. Highlights of Progress*

Atomic level determination of the structure of cytochromeb562. During the past year we greatly expanded the functionality of PROTEAN towards obtaining meaningful biochemical results. In the first half of the year the primary outcome of this effort was the atomic-level assembly of the protein cytochromeb562 using simulated NMR constraints from the known crystal structure. The strategy for assembling cytochrome involved functionality at the atomic as well as the more abstract solid level, and demonstrated one possible application of our general approach, which we call *heuristic refinement.*

The heuristic refinement method falls within an *exclusion paradigm* for interpreting protein structure. In this paradigm all the atoms in the protein are initially assumed to be everywhere with respect to some arbitrary coordinate system. As constraints are introduced these initially infinite *accessible volumes* are gradually reduced until the remaining accessible volumes are small enough so that a representative set of structures may be enumerated.

The exclusion paradigm has not been tried previously because it appears to be computationally intractable for any reasonably sized protein. For this reason most current techniques (non AI-based) use an *adjustment paradigm* to optimize a single structure towards a global minimum. These techniques all share difficulties common to optimization, namely lack of convergence for large numbers of variables or for starting structures far from the global minimum. In addition, for reasons of computation time, they are not able to provide a representative sample of all structures compatible with the constraints.

The exclusion paradigm, by systematically eliminating structures incompatible with the constraints, has the advantage that all, or at least a representative set, of structures compatible with the constraints are retained. In order to deal with the combinatorics the heuristic refinement method formulates the problem as one of search, which is the fundamental paradigm of artificial intelligence. In particular, the problem is formulated as a geometric constraint satisfaction problem (GCSP), which can be solved by backtrack search if the size of the atomic accessible volumes can be made small enough.

The complexity of solving a GCSP depends on the number of objects and the number of possible locations that each object can have in space. Therefore, the heuristic refinement method utilizes four techniques to reduce these two numbers to the point that backtrack search may be applied. The effect of each of these techniques is to utilize knowledge of protein structure and of techniques for solving GCSPs to prune the search tree:

1. Problem decomposition. The protein is broken into logical subparts such as secondary structures and sidechains. Each of these is treated as a separate GCSP, which is partially solved. Solutions to the subproblems either determine constraints at more abstract levels or are combined into larger subproblems.

2. Problem abstraction. Groups of locally highly constrained atoms are represented as single abstract "solid" level objects. Solutions to atomic level GCSPs are used to create abstract "solid" level constraints, resulting in a solid level GCSP with far fewer objects. Solutions at the solid level are then expanded to the atomic level, resulting in fewer possible locations for the atoms than would have been present if the initial solid level processing had not occurred. Almost all the work reported in previous reports was at this abstract solid level, but it is the current refinement to the atomic level that is of most interest to biochemists.

3. Local satisfaction of constraints. In each local GCSP filtering operations called network consistency algorithms are applied, allowing the accessible volumes to be reduced by looking at constraints pairwise (or to higher order) rather than all at once. These operations, which were initially developed for computer vision, are becoming more popular in AI because of their utility in many forms of constraint satisfaction problems.

4. Heuristic control. At each point in the problem solving there are many possible constraint satisfaction operations that may be applied. If there is no bias then the order of operations should not matter, but the efficiency may vary greatly. Much of the previously reported work in PROTEAN went towards developing the BB1 framework for using heuristics to opportunistically determine the best operation to apply at any given time. The BB1 framework has been used to control the solid abstract level of problem solving, but it has not yet been integrated with the atomic level functionality, which is currently manually controlled.

These principals were used to determine the structure of cytochromeb562. The program was given as input the primary and secondary structure as obtained from the known crystal structure. It was also given a set of 729 distance ranges designed to simulate the expected type of data from NMR. Using these data, as well as knowledge of protein structure, the heuristic refinement method produced a set of 10 atomic level structures, with an average root mean square deviation from the crystal structure of 4.1 angstroms when all backbone alpha-carbons were included, and 2.8 angstroms when random-coil segments were excluded. These deviations were on the order of those found by other methods. Because we systematically excluded structures we could be confident that we had obtained an upper bound on the set of structures compatible with the constraints. These structures could then be adjusted by any of the current techniques (although we didn't do that in this case).

The cytochrome results were obtained over a period of about three to four months, using the previously-described geometry system and BB1 to obtain the solid level results, and LISP and C code to produce the atomic level results. Intermediate results were written to files, and individual programs were written by three different people.

Removing bias. During the latter part of this year we have concentrated on improving the functionality used to obtain the cytochrome results.

In the heuristic refinement method an important tradeoff is efficiency versus the amount of bias introduced by the abstractions. A major source of bias is the assumption that secondary structures are in their ideal configuration. For example, in the cytochrome and all previously reported results alpha helices were modeled as cylinders. This approximation meant that some legal atomic level solutions are prematurely eliminated at the solid level. In order to reduce this bias we have looked at actual examples of helices in the crystal structure database, and have used these to form more realistic constraints at the solid level. This relaxation of the ideal helix assumption has allowed us to obtain more accurate solid level accessible volumes, but at the cost of less constraining power. In later versions of the program we plan to allow the user to specify how much ideality to assume.

Atomic level constraint satisfaction operations. As part of his Ph.D. work Bruce Duncan has extended many of the solid level constraint satisfaction operations to the atomic level. He has developed programs, written in C, which allow atomic locations to be described at different levels of resolution, from a coarse grid initially to a finer grid as refinement proceeds. This approach has been used to reconstruct the crystal structure of a small protein called crambin given exact distances less than five angstroms. In this case no solid level preprocessing was done, and the entire process took about six weeks on three Suns running in parallel. Bruce is currently working to develop more intelligent control before trying his system on a more realistic simulated data set.

Probabilistic operations at the atomic level. Russ Altman, as part of his Ph.D. thesis, has been working to develop an alternative method for representing the accessible volumes of atoms. Currently, we represent accessible volumes as lists of xyz locations sampled on a regular grid at a specified resolution. In Russ's approach accessible volumes are represented as probability density functions described by a mean and covariance matrix. Constraints are also represented by a mean and variance, and a Kalman filter, which is basically another form of Bayes Formula, is used to determine the reduction in accessible volume. This approach has been used to reconstruct the amino acid tyrosine, and is being extended to handle larger numbers of atoms.

Development of an integrated software environment and general strategies. All our current results were obtained with more or less independently developed software modules. No one person is able to run PROTEAN by himself. For this reason we have designed a framework that allows these modules to communicate via common files organized as a distributed object-oriented knowledge base. Parts of this framework are currently being implemented by a part time programmer. The framework will allow the existing solid level functionality to be integrated with the emerging atomic level, and will allow different assumptions, representations and algorithms to be experimented with.

Porting of heuristic control to TI Explorer in Common Lisp. The reasoning component in BB1 is currently written in InterLisp and runs on Xerox D-machines. Since InterLisp and D-machines are becoming obsolete the code has now been partially ported to the TI Explorer in Common Lisp. Network software has been written that allows the Explorer to control the geometry and graphics programs over the network. Once the port has been completed additional heuristic control experiments will be performed.

Secondary structure prediction. An important pre-processing step for PROTEAN is the determination of secondary structure from NMR. John Brugge, for his masters thesis, designed a program called ABC which emulates expert knowledge in assigning secondary structure from NMR. The program is written in BB1 and uses symbolic patterns in the NMR to predict structure. Validation studies showed that the program did about as well as experts.

Peak Assignment. When an adequate amount of purified protein is available, the *routine* determination of protein structure in solution by nuclear magnetic resonance would require efficient methods for: 1) collecting data, 2) extracting structural constraints from the data, and 3) generating protein structures which satisfy these constraints. Many two-dimensional nuclear magnetic resonance (2D-NMR) experiments now exist to carry out step 1), and a number of computer methods have been presented to deal with step 3). However, step 2) is a severe bottleneck.

The main difficulty in that step lies in the assignment problem in which each peak from 2D-NMR spectra must be matched to a pair of specific atoms from specific residues. Typically, hundreds of resonances must be identified, a tedious, slow, and error prone task. It involves keeping track of a large number of peaks, each of which initially has a large number of possible assignments. This complexity, however, contrasts with the relatively small number of principles and pattern matching techniques that are actually needed to assign the peaks in a 2D-NMR protein spectra.

Craig Cornelius, Guido Haymann-Haber, and Olivier Lichtarge have developed a prototype program called PEAKS that uses constraint satisfaction methods to apply patterns expected from amino acid residues to actual spectra. The program has been tested on simulated data sets for small proteins of 20 to 30 residues. The side chains of most residues can be correctly identified and sometimes assigned. The program can also deal with slightly incomplete spectra with degenerate or missing peaks.

## *D. Relevant Publications*

1. Altman, R. and Jardetzky, O.: *New strategies for the determination of macromolecular structures in solution.* Journal of Biochemistry (Tokyo), Vol. 100, No. 6, p. 1403-1423, 1986.

2. Altman, R. and Buchanan, B.G.: *Partial Compilation of Control Knowledge.* Proceedings of the AAAI, pp 399-404, 1987.

3. Altman, R., Duncan, B., Brinkley, J., Buchanan, B., Jardetzky, O.: *Determination of the spatial distribution of protein structure using solution data,* Proceedings of the Alfred Benzon Symposium 26: NMR Spectroscopy and Drug Development, Copenhagen, 1987.

4. Brinkley, J., Cornelius, C., Altman, R., Hayes-Roth, B. Lichtarge, O., Duncan, B., Buchanan, B.G., Jardetzky, O.: *Application of Constraint Satisfaction Techniques to the Determination of Protein Tertiary Structure.* Report KSL-86-28, Department of Computer Science, 1986.

5. Brinkley, James F., Buchanan, Bruce G., Altman, Russ B., Duncan, Bruce S., Cornelius, Craig W.: *A Heuristic Refinement Method for Spatial Constraint Satisfaction Problems.* Report KSL 87-05, Department of Computer Science.

6. Brinkley, J.F., Altman, R.B., Duncan, B.S., Buchanan, B.G., and Jardetzky, O.: *The heuristic refinement method for the derivation of protein solution structures: Validation on cytochrome-b562,* KSL Technical Report 88-03, 1988.

7. Brugge, J.A., Buchanan, B.G., and Jardetzky, O.: *Toward automating the process of determining polypeptide secondary structure from $^1H$ NMR data,* To be published in J. Computational Chemistry, 1988.

8. Buchanan, B.G., Hayes-Roth, B., Lichtarge, O., Altman, A., Brinkley, J., Hewett, M., Cornelius, C., Duncan, B., Jardetzky, O.:*The Heuristic Refinement Method for Deriving Solution Structures of Proteins.* Report KSL-85-41. October 1985.

9. Duncan, B., Buchanan, B., Hayes-Roth, B., Lichtarge, O., Altman, R., Brinkley, J., Hewett, M., Cornelius, C., and Jardetzky, O.: *PROTEAN: A new method for deriving solution structures of proteins,* Bull. Mag. Res., 8:111-119, 1987.

10. Garvey, Alan, Cornelius, Craig, and Hayes-Roth, Barbara: *Computational Costs versus Benefits of Control Reasoning.* Report KSL 87-11, Department of Computer Science.

11. Hayes-Roth, B.: *The Blackboard Architecture: A General Framework for Problem Solving?* Report HPP-83-30, Department of Computer Science, Stanford University, 1983.

12. Hayes-Roth, B.: *BB1: An Environment for Building Blackboard Systems that Control, Explain, and Learn about their own Behavior.* Report HPP-84-16, Department of Computer Science, Stanford University, 1984.

13. Hayes-Roth, B.: *A Blackboard Architecture for Control.* Artificial Intelligence 26:251-321, 1985.

14. Hayes-Roth, B. and Hewett, M.: *Learning Control Heuristics in BB1.* Report HPP-85-2, Department of Computer Science, 1985.

15. Hayes-Roth, B., Buchanan, B.G., Lichtarge, O., Hewett, M., Altman, R., Brinkley, J., Cornelius, C., Duncan, B., and Jardetzky, O.: *PROTEAN: Deriving protein structure from constraints.* Proceedings of the AAAI, 1986, p. 904-909.

16. Jardetzky, O.: *A Method for the Definition of the Solution Structure of Proteins from NMR and Other Physical Measurements: The LAC-Repressor Headpiece.* Proceedings of the International Conference on the Frontiers of Biochemistry and Molecular Biology, Alma Alta, June 17-24, 1984, October, 1984.

17. Lichtarge, Olivier: *Structure determination of proteins in solution by NMR.* Ph.D. Thesis, Stanford University, November, 1986.

18. Lichtarge, Olivier, Cornelius, Craig W., Buchanan, Bruce G., Jardetzky, Oleg: *Validation of the First Step of the Heuristic Refinement Method for the Derivation of Solution Structures of Proteins from NMR Data.*, Proteins: Structure, Function and Genetics, 2:340-358, 1987.

### *E. Funding Support*

Title: Interpretation of NMR Data from Proteins Using AI Methods
PI's: Oleg Jardetzky and Bruce G. Buchanan
Agency: National Science Foundation
Grant identification number: DMB-8402348
Total Award Period and Amount: 2/1/87 - 9/30/89 $120,000
(includes direct and indirect costs)
Current award period and amount: 2/1/87 - 9/30/89 $120,000
(includes direct and indirect costs)

The following grants and contracts each provide partial funding for PROTEAN personnel.

Title: Knowledge-Based Systems Research
PI: Edward A. Feigenbaum
Agency: Defense Advanced Projects Research Agency
Grant identification number: N00039-86-0033
Total award period and amount: 10/1/85 - 9/30/88 $4,130,230
(direct and indirect)
Current award period and amount: 10/1/87 - 9/30/88 $1,467,300
(direct and indirect)
PROTEAN component is $72,956, or 5.0 % of grant total

## II. INTERACTIONS WITH THE SUMEX-AIM RESOURCE

### *A. Medical Collaborations*

Several members of Prof. Jardetzky's research group are involved in this research.

### *B. Interactions with other SUMEX-AIM projects*

We are occasionally in contact with researchers at Robert Langridge's laboratory at the University of San Francisco.

### *C. Critique of Resource Management*

The SUMEX staff has continued to be most cooperative in supporting PROTEAN research. The SUMEX computer facility is well maintained and managed for effective support of our work. The computer network and Lisp workstations are supported very effectively by the SUMEX staff.

## III. RESEARCH PLANS

### *A. Goals & Plans*

Our long range biochemical goal is to build a system that can automatically interpret NMR and other types of constraints, in order to produce the family of structures compatible with the constraints. In so doing we have encountered and will continue to encounter important and interesting artificial intelligence problems in large scale, knowledge-based constraint satisfaction.

The major steps towards these goals are as follows, more or less in order of expected order, although the validations will proceed in parallel with the addition of new features. Several of these steps have been worked on within the past year (see Highlights of Research Progress).

*A.1. Develop an integrated system.* In order for the heuristic refinement method to be widely used it must be integrated into a computer system that contains the required functionality, is reasonably efficient, is semi-automatically controlled, relatively easy to use, and extensible. Using our previous work as a basis we will continue to work towards this goal. The required substeps in order to achieve the goal of an integrated system are:

- Develop improved functionality. This is our current major focus and includes efforts to relax the ideal helix assumption, to utilize proven techniques from other structure determination methods (such as the use of distance bound smoothing algorithms as a pre-processing step, and the use of adjustment procedures as a post-processing step), to develop better methods of representing the spatial locations of both three-dimensional (non-oriented) objects, and six-dimensional (oriented) objects, and to develop more efficient constraint satisfaction algorithms. Progress has been made in several of these areas over the past year, especially by the Ph.D. students as they develop their dissertations.

- Develop system architecture. During this year we have also designed a basic architecture for a distributed system that can be extended as new modules and representations are developed. Much of the programmers effort this year will go towards implementing this system and towards incorporating some of the advances made by the Ph.D. students.

- Develop control strategies and knowledge base. As the integrated functionality is developed it will be made available either as remote servers accessible over the network (as with our current geometry system), or as loadable modules. BB1, or another suitable object-oriented expert system shell, will be used to create the control modules. These modules will use heuristics to intelligently pick the most efficient constraint satisfaction operation(s) to perform next. Progress towards one such control module has been made with the porting of the BB1 system to the TI Explorer.

- Develop user interface and improve graphics. To be useful the system should be relatively easy to run by a non-programmer biochemist. We have not yet begun to explore this issue, but other AI in Medicine projects at Stanford and elsewhere have worked in this area. Examples at Stanford are Oncocin and its derivatives, and Apple's HyperCard as an interface to an object-oriented belief network. Once the programmer has created a rudimentary and extensible system he will begin to look at this issue.

*A.2. Extend the system.* As the required functionality, framework and control strategies become integrated a version of the system will be made available (at least at Stanford) for analyzing real data. At the same time we will use the integrated system as the basis for extending its capabilities. Developing these new capabilities will require solution to some important AI problems dealing with object representation and reasoning over time as well as space. These extensions include:

- Studying larger proteins and protein complexes. Since our system is hierarchical there is in theory no reason why we can't move all the way up the anatomic spectrum from atoms to organs.
- Add new types of constraints, including global constraints such as volume and surface, and probabilistic criteria. The current work of Russ Altman should be very useful in this regard.
- Automate the earlier interpretation of NMR data. We have made good beginnings in this area with the ABC program for secondary structure prediction and the peak detection programs. We will also continue our development of the PEAKS system for interpretation of 2-D NMR spectra, to be integrated with ABC and the rest of the PROTEAN system.
- Analyze structure. The purpose of determining structure is to understand function, and a component of this is analyzing the structures produced. We have already developed several modules for analyzing structure, such as programs for determining root mean square deviations and for calculating distance distributions. Russ Altman has also done some work in automatically extracting features of interest from a protein, and will continue this as part of his dissertation.
- Study protein dynamics. In some cases proteins in solution may be moving, a fact which greatly complicates the analysis of NMR data. So far we have ignored this issue because it is hard enough to solve the problem with stationary structures, but as we gain experience we will begin to consider these issues. Many problems dealing with incompatible constraints will arise as we look at this problem.

*A.3. Validate on known crystal structures.* As new functionality is developed, it will be verified by attempting to reconstruct proteins whose structures are known from crystallography. In this way we will at least know that our methods work, even if there is no gold standard for solution data.

*A.4. Use on actual NMR and theoretical constraints.* As the system is developed and verified on crystal data it will be made available to biochemists. Only those capabilities that have been verified will be supported. Funds for actually using the system on real data will be expected to be provided in the context of specific research projects.

### *B. Justification for continued SUMEX use*

We will continue to use the SUMEX facilities for integrating and expanding the PROTEAN system. The 2060 (or whatever replaces it) will primarily be a focus for integration and communication among machines and humans. Most of the actual program development will be on workstations running C or Common Lisp, communicating via the excellent local area network maintained by SUMEX. The communication and clearinghouse capability provided by SUMEX and the SUMEX staff is essential for continued success of such a large and evolving distributed project.

### *C. Need for other computing resources*

At this time our computational resources are almost adequate as long as we can continue to use the SUMEX-supported workstations and local area network. We could always use faster machines since many of our computations are very numerically intensive, but our major need is for personnel and systems support. Since PROTEAN is evolving into a distributed problem-solving system we actively support and require extensive networking and remote procedure access capabilities. We believe that the requirements of PROTEAN make it an ideal test problem for many of the distributed systems issues being considered by the SUMEX core research staff.

# IV.A.6. RADIX and PENGUIN Projects

**The RADIX Project: Deriving Medical Knowledge from Time-Oriented Clinical Databases**

**The PENGUIN Project: Applying Database and Knowledge base Technology to Medical Instrumentation**

**Gio Wiederhold, Ph.D.**
**Departments of Computer Science and Medicine**
**Stanford University**

**Thierry Barsalou, M.D.**
**Medical Computer Science**
**Stanford University**

**Robert L. Blum, M.D., Ph.D.**
**Department of Computer Science**
**Stanford University**

## I. SUMMARY OF RESEARCH PROGRAM

The RADIX research has been phased out during this year. Although proposals to NLM and NCHSR for continued research support were approved, they have not been funded. Some closeout funding enabled the project to be brought to an orderly conclusion, and publications continue to appear. The experience gained here and under the predecessor project, RX, continues to influence basic research directions. The problem of automated knowledge acquisition remains an important area of research, and we expect that the foundations laid here will influence work of others as well.

Some of the basic research has influenced the KBMS/KSYS project. Specifically problems encountered in the management and structuring of large quantities of knowledge are now being addressed, primarily under DARPA sponsorship. A subproject, on knowledge validation is being supported by IBM and will draw directly on the data and knowledge bases established for RADIX in its initial phases.

The KBMS/KSYS work, in turn, is leading to another important medical application: PENGUIN. This project is currently mainly supported by KBMS, but a substantial research grant has been approved by NLM, and given a high priority, so that we are confident that we will be able to soon move forward intensely in this project.

We will, below, provide two distinct sections. The first one will provide the final report of achievements under RADIX and the second will outline the current state and plans for PENGUIN.

## RADIX PROJECT GOALS AND PROGRESS

### *A. RADIX -- Technical Goals*

The objectives of the RADIX project were 1) Discovery: to provide knowledgeable assistance to a research investigator in studying medical hypotheses on large databases, and to automate the process of hypothesis generation and exploratory confirmation, 2) Summarization: to develop a program and set of techniques for automated summarization of patient records, and 3) Peer Review: to develop a program to assist physician reviewers examine case databases for medical peer review and quality assurance. For system development we have used a subset of the ARAMIS database. We will first describe our work on discovery, followed by summarization and peer review. Research was performed on the first two objectives.

*RADIX Discovery Module*: Computerized clinical databases and automated medical records systems are starting to contain valuable long-term records of medical practice and experience. We have utilized the data collected by the ARAMIS Project, (American Rheumatism Association Medical Information System) under development since 1969 in the Stanford Department of Medicine. ARAMIS contains records of over 17,000 patients with a variety of rheumatologic diagnoses. Over 62,000 patient visits have been recorded, accounting for 50,000 patient-years of observation.

A fundamental objective of the ARAMIS Project and many other clinical database projects is to use the data that have been gathered by clinical observation in order to study the evolution and medical management of chronic diseases. Unfortunately, the process of reliably deriving knowledge has proven to be exceedingly difficult. Numerous problems arise stemming from the complexity of disease, therapy, and outcome definitions, from the complexity of causal relationships, from errors introduced by bias, and from frequently missing and outlying data. A major objective of the RADIX Project is to explore the utility of symbolic computational methods and knowledge-based techniques at solving some of these problems.

The first phase of the RADIX computer program helps to examine the time-oriented ARAMIS database displays information which helps recognize possibly causal relationships. The important concept is here the mapping of base information to higher level abstractions which are meaningful to physicians. Instances of data, when mapped, can display a patient's course in a compact and meaningful fashion.

The intent was also that knowledge acquisition from physician-researchers can then be brought to a level where the relationships are meaningful. Instances of such relationships were acquired in the RX projects, but that work was not yet based on the more general concepts envisaged for RADIX.

*RADIX Summarization Module*: The management of inpatients and outpatients is often complicated by the size and disorganization of patient charts. Computerized patient records are becoming increasingly available. While computerization of records renders data, it worsens the problem of information overload. The ability to automatically create patient summaries represents a useful adjunct to a patient record for rapid review of a case, for clinical decision making and patient monitoring, and for surveillance of quality of care. The goal of the RADIX summarization program is to infer a summary of a patient's clinical history from lengthy on-line medical records.

The RADIX summarization program is a knowledge-based sub-system which produces intelligent summaries from a time-oriented data base of Systemic Lupus Erythematosus patients. Medical concepts in the system are represented by three entities of increasing complexity: abnormal primary attributes, abnormal states and diseases. Abnormal states and diseases are derived from the abnormal primary attributes by the Reasoner using a combination of model-driven and data-driven algorithms. Uncertainty associated with the derived states is handled with a Bayesian approach supplemented by boolean predicates, using likelihood ratios obtained from a transformation of the INTERNIST knowledge base. After summarizing the data, the system generates interactive, graphical displays with optional explanation windows.

The prototypes we have implemented have shown that intelligent summarization of medical records is feasible and that interactive graphical display is of great help in conveying complex medical information. We have reported on the results of this work.

### *B. RADIX -- Medical Relevance and Collaboration*

As a test bed for system development, our focus of attention were the records of patients with systemic lupus erythematosus (SLE) contained in the Stanford portion of the ARAMIS Data Bank. SLE is a chronic rheumatologic disease with a broad spectrum of manifestations. Occasionally the disease can cause profound renal failure and lead to an early death. With many perplexing diagnostic and therapeutic dilemmas, it is a disease of considerable medical interest.

The medical relevance of the automated summarization program is readily apparent. A practicing physician or medical researcher, faced with a patient chart, often with dozens of visits and scores of attributes, rarely has time to read the entire chart. He (or she) would like a succinct summary of the important events in that patient's record to assist his decision making. The use of computerized medical records improves the quality of information but does not solve the problem of information overload. For this reason, it would be useful to have the ability to automatically summarize patient records into meaningful clinical events.

### *C. RADIX -- Highlights of Research Progress*

#### *C.1 April 1987 to September 1988*

Our primary accomplishments in this period have been the following:

1. Implementation and demonstration of a second generation of the automated summarization program.
2. Application of algorithms for transforming the Internist knowledge base into standard Bayes form.
3. Publication of papers on automated summarization, and presentation of results at medical conferences.
4. Training post-doctoral researchers, participants in RADIX, in methods of medical artificial intelligence research.

#### *C.1.1 Design and implementation of a second generation of the prototype automated summarization program*

We have completed a second generation of our prototype automated summarization program. This work is described in DeZegher-Geets, 1987, noted in the publications section. It improves upon a prototype implemented by Downs (Downs 1986); the knowledge base has been substantially enlarged, the inference mechanisms refined and enhanced for temporal reasoning, and the graphical display capability has been expanded. The summarization program produces intelligent summaries from a time-oriented data base of Systemic Lupus Erythematosus patients. Medical concepts in the system are represented by three entities of increasing complexity: abnormal primary attributes, abnormal states and diseases. Abnormal states and diseases are derived from the abnormal primary attributes by the Reasoner using a combination of model-driven and data-driven algorithms. Uncertainty associated with the derived states is handled with a Bayesian approach supplemented by boolean predicates, using likelihood ratios obtained from a transformation of the INTERNIST knowledge base. After summarizing the data, the system generates interactive, graphical displays with optional explanation windows.

### *C.1.2 Algorithms for transforming the Internist knowledge base into standard Bayes form*

INTERNIST-1 is an expert system for diagnosis across a broad spectrum of disease. Over twenty man-years of effort have gone into the construction of its knowledge base which contains relationships between approximately 600 diseases and 4,000 manifestations of disease. A major limitation of INTERNIST-1 is that the quantities used within the system to represent uncertainty, called evoking strengths and frequencies, are poorly defined. This makes it difficult to tune the method used by the program to assign likelihoods to diseases (the scoring scheme) and makes it difficult to transport knowledge contained in the program to other medical diagnostic systems.

In collaboration with R. Miller and D. Heckerman we have transformed the values in the INTERNIST KB into a probabilistic form. Various combinations of multiple regressions were performed on the evoking strengths, frequencies, probabilities of disease, p(D), and probabilities of manifestation, p(M), versus the likelihood ratios L(D|M) and L(D|not M). This process yielded some interesting and unexpected results. For example, the multiple regression of evoking strength AND p(M) vs. L(D|M) showed an r-squared of 0.84, significantly better than the r-squared value for evoking strength vs. L(D|M) alone. Also, the transformation from frequency, p(M), and p(D) into L(D|not M) revealed a correlation coefficient of 0.58. These results suggest a low cost method for converting the knowledge in INTERNIST-1 to a probabilistic form. In particular, assessments of p(D) and p(M) (only about 4500 numbers) can be used in conjunction with evoking strengths and frequencies in the KB (about 40,000 numbers) to construct likelihood ratios.

### *C.1.3 Publication of papers on automated summarization, and presentation of results at medical conferences*

We have submitted and had accepted several papers, as noted in the section on publications. Some of these were presented at medical conferences.

### *C.1.4 Training Post-Doctoral researchers, participants in RADIX, in methods of medical artificial intelligence research*

We have been training three post-doctoral researchers on the project during the earlier part of the current reporting year; Andrew G. Freeman, M.D., Isabelle de Zegher-Geets, M.D., and Donald Rucker, M.D.. Andrew Freeman has developed the Internist transformation algorithms. Isabelle de Zegher-Geets completed a thesis on Automated Summarization as part of Stanford's Medical Information Sciences program.

## PENGUIN PROJECT GOALS AND PROGRESS

### *A. PENGUIN Project -- Rationale*

Databases and expert systems share a common goal -- generating useful information for action -- but accomplish their tasks separately, using different principles. It is clear, however, that future information systems will require both the problem-solving capabilities of expert systems (ESs) and the data-handling capabilities of database management systems (DBMSs). Indeed, combining database and expert system technologies into expert database systems (EDSs) is an emerging research area. One can define an EDS as "a system for developing applications requiring knowledge-directed processing of shared information". From a perspective of developing advanced biomedical information systems, this definition conveys two precise scenarios: (1) enhancing DBMSs with structuring and manipulation tools that take more semantics into account; (2) allowing ESs to access and to handle efficiently information stored in database(s).

The object-oriented paradigm has gained much attention in recent years. In the database field, object-oriented DBMSs have emerged. The concept of an entity is also widely used by database design tools. In the field of artificial intelligence, frames are a well-known knowledge representation scheme. Although frames were conceived separately from the object paradigm, the two are in fact consistent with each other. In this research project, called PENGUIN, we investigate the hypothesis that the object-oriented approach can also serve as a unifying scheme for developing EDSs.

We have the opportunity to explore this hypothesis in a practical biomedical environment. Fluorescence Activated Cell Sorting (FACS) is emerging as a major source of information for biomedical research and clinical practice. Currently, achieving good FACS performance requires analyzing and integrating various complex data and knowledge sources. PENGUIN is thus aimed at developing methods for bringing together expert system and database technologies in an integrated advice system that could fulfill the information needs of FACS investigators. Central to this work is a very close collaboration between researchers in the Medical Information Sciences Program and the Departments of Computer Science and Genetics.

### *B. PENGUIN -- Medical Relevance and Collaboration*

FACS has already demonstrated significant promise in clinical as well as research-oriented areas. As indicated earlier, basic studies identifying lymphocyte subpopulations have now been translated into clinical applications including monitoring of AIDS patients. Similar FACS applications can now be found in such diverse specialties as Hematology, Oncology, Infectious Diseases and even Gynecology-Obstetrics where current studies are evaluating FACS analysis of maternal blood sample as a potential non-invasive method for prenatal diagnoses.

The major block to proliferation of these kinds of valuable studies involves the requirement for greater skills in reagent selection and FACS machine operation than are typically available in basic and clinical research settings.

Thus developing sophisticated computer tools that provide a new level of automatic analysis and control for FACS and greatly facilitates FACS use by reducing the need for on-site human expertise should significantly improve the potential for using this versatile methodology in biomedical research and clinical practice.

### *C. PENGUIN -- Highlights of Research Progress*

Our current effort focuses on developing a computer-based advisory system to assist the FACS investigator in designing experiment protocols. As mentioned above, this system will combine database and artificial intelligence methodologies in an integrated framework, as to provide 1) several levels of abstraction for information management and retrieval from a relational database system, 2) access to and integration of the results of past similar experiments as additional units of information through an interface with existing databases of FACS data and 3) inference capabilities coupled to the database for high-level interactions with scientists during the design process. Although this work is motivated by a specific application, the formal design of the system will be domain-independent; it is then our belief that ideas, principles and programs developed in this process will be applicable to other medical and non-medical areas.

In the past year, we have developed the concept of an object-based interface on top of a relational database system and implemented an initial prototype of the interface. We have drawn an analogy between the notions of object and database view. Using this analogy, we have defined three components in the object interface:

1. The object *generator* maps relations into object templates where each template can be a complex combination of join (combining two relations through shared attributes) and projection (restricting the set of attributes of a relation) operations on the base relations. In addition, an object network groups together related templates, thereby identifying different object views of the same database. The whole process is knowledge-driven, using the semantics of the database structure. We define the object schema as the set of object networks constructed over a given database. Like the data schema for a relational database, the object schema represents the domain-specific information needed to gain access to PENGUIN's objects; this information enables us to combine well-organized, regular tabular structures -- the relations -- into complex, heterogeneous entities -- the objects.

2. The object *instantiator* provides nonprocedural access to the actual object instances. First, a declarative query (e.g., Select instances of template x where attribute y < 0.5) specifies the template of interest. Combining the database-access function (stored in the template), and the specific selection criteria, PENGUIN automatically generates the relational query and transmits it to the DBMS, which in turn transmits back the set of matching relational tuples. In addition to performing the database-access function, the object template specifies the structure and linkage of the data elements within the object. This information is necessary for the tuples to be correctly assembled into the desired instances. Those instances are then made available to the expert system directly, or to the user through a graphic interface.

3. The object *decomposer* implements the inverse function; that is, it maps the object instances back to the base relations. This component is invoked when changes to some object instances (e.g., deletion of an instance, update of some attributes) need to be made persistent at the database level. An object instance is generated by collapsing (potentially) many tuples from several relations. By the same token, one update operation on an object may result in a number of update operations that need to be performed on the base relations.

Preliminary results of PENGUIN's implementation indicate that such an object-based architecture provides (1) efficient access to and manipulation of complex units of biomedical information while preserving the advantages associated with persistent storage of data in relational format, and (2) a domain-independent, bidirectional communication channel between relational database systems and expert systems.

In parallel to developing the object interface, we are currently engaged in a knowledge acquisition process to define the structure of the database and of the knowledge base. Through interviews with our experts in the Genetics department, we are eliciting the structure of the various components (genetic, histological, and serological information) of the FACS reagents database. Finally, we are exploring the use of hypertext (more specifically, the HyperCard program for the Macintosh) as a possible framework for developing the user interface component of PENGUIN.

*D. Publications of the RADIX and PENGUIN projects*

1. Barsalou, Thierry and Gio Wiederhold: "Automating a Cell Counter"; to be published in The International Journal of Artificial Intelligence in Engineering, Computational Mechanics Publ., UK, 1988.

2. Barsalou, Thierry and Gio Wiederhold: "Applying a Semantic Model to an Immunology Database"; in W.W. Stead (editor), Proceedings of the Eleventh Symposium on Computer Applications in Medical Care, pages 871-877, IEEE Computer Society Press, Washington, D.C., November 1987.

3. Barsalou, Thierry, W.A. Moore, L.A. Herzenberg, and G. Wiederhold: "A Database System to Facilitate the Design of FACS Experiment Protocols" (abstract); Cytometry, Vol.97, August 1987.

4. Barsalou, T. and G. Wiederhold. *Knowledge-based mapping of relations into objects*. To appear in the International Journal of AI in Engineering, 1988.

5. Barsalou, T. *An object-based architecture for biomedical expert database systems*. Submitted to the IEEE Twelfth Symposium on Computer Applications in Medical Care, 1988.

6. Blum, Robert L, Computers and Artificial Intelligence in Clinical Medicine: Current Systems and Future Prospects, Computer News for Physicians, October, 1987.

7. Blum, R. L., and Walker, M.G.: *LISP as an Environment for Software Design: Powerful and Perspicuous.* In Proceedings of the Tenth Annual Symposium on Computer Applications in Medical Care, pages 326-331. IEEE Computer Society, October, 1986.

8. Blum, R. L., and Walker, M.G.: *Automated Medical Discovery from Clinical Databases: An Overview of the RADIX Project.* In Proceedings of the Fifth Toyobo Biotechnology Foundation Symposium: Artificial Intelligence in Medicine. The Toyobo Foundation, Tokyo, Japan, August, 1986.

9. Blum, R. L., and Walker, M. G.: *Automated Medical Discovery from Clinical Databases: an Overview of the RADIX Project.* In Proceedings of the First International Conference on Artificial Intelligence and Its Impacts in Biology and Medicine, pages 59-83. Groupement Scientifique pour le Developpement de l'Intelligence Artificielle en Languedoc-Roussillon, Montpellier, France, September, 1986.

10. Blum, Robert L. and Gio C.M. Wiederhold: *Studying Hypotheses on a Time-Oriented Clinical Database: An Overview of the RX Project.* In J.A. Reggia and S. Thurim: 'Computer Assisted Medical Decision-Making'; Springer Verlag, 1985, pp.245-253.

11. Blum, R.L.: *Two Stage Regression: Application to a Time-Oriented Clinical Database.* Knowledge Systems Laboratory Technical Report. 1985.

12. Blum, R.L.: *Modeling and encoding clinical causal relationships.* Proceedings of SCAMC, Baltimore, MD, October, 1983.

13. Blum, R.L.: *Representation of empirically derived causal relationships.* IJCAI, Karlsruhe, West Germany, August, 1983.

14. Blum, R.L.: *Machine representation of clinical causal relationships.* MEDINFO 83, Amsterdam, August, 1983.

15. Blum, R.L.: *Clinical decision making aboard the Starship Enterprise.* Chairman's paper, Session on Artificial Intelligence and Clinical Decision Making, AAMSI, San Francisco, May, 1983.

16. Blum, R.L. and Wiederhold, G.: *Studying hypotheses on a time-oriented database: An overview of the RX project.* Proc. Sixth SCAMC, IEEE, Washington D.C., October, 1982.

17. Blum, R.L.: *Induction of causal relationships from a time-oriented clinical database: An overview of the RX project.* Proc. AAAI, Pittsburgh, August, 1982.

18. Blum, R.L.: *Automated induction of causal relationships from a time-oriented clinical database: The RX project.* Proc. AMIA San Francisco, 1982.

19. Blum, R.L.: *Discovery and Representation of Causal Relationships from a Large Time-oriented Clinical Database: The RX Project.* In D.A.B. Lindberg and P.L. Reichertz (Eds.), LECTURE NOTES IN MEDICAL INFORMATICS, Springer-Verlag, 1982.

20. Blum, R.L.: *Discovery, confirmation, and incorporation of causal relationships from a large time-oriented clinical database: The RX project.* Computers and Biomed. Res. 15(2):164-187, April, 1982.

21. Blum, R.L.: *Discovery and representation of causal relationships from a large time-oriented clinical database: The RX project* (Ph.D. thesis). Computer Science and Biostatistics, Stanford University, 1982.

22. Blum, R.L.: *Displaying clinical data from a time-oriented database.* Computers in Biol. and Med. 11(4):197-210, 1981.

23. Blum, R.L.: *Automating the study of clinical hypotheses on a time-oriented database: The RX project.* Proc. MEDINFO 80, Tokyo, October, 1980, pp. 456-460. (Also STAN-CS-79-816)

24. Blum, R.L. and Wiederhold, G.: *Inferring knowledge from clinical data banks utilizing techniques from artificial intelligence.* Proc. Second SCAMC, IEEE, Washington, D.C., November, 1978.

25. DeLaPaz, R., Hanson, W., Bernstein, R., and Walker, M.G. "Tissue Characterization of MRI Using Fuzzy C-means Cluster Analysis" In preparation for Radiology.

26. DeLaPaz, R., Hanson, W., Bernstein, R., and Walker, M.G. "Clinical Application of Automated MRI Tissue Classification" In preparation for Magnetic Resonance in Medicine.

27. DeLaPaz, R., Bernstein, R., Chang, P., Hanson, W., and Walker, M.G.

"Approximate Fuzzy C-means (AFCM) Cluster Analysis of Medical Magnetic Resonance Image (MRI) Data" Submitted to the International Conference on Pattern Recognition, Special Session on Diagnostic Medical Image Processing, Beijing, China, October 1988.

28. DeZegher-Geets, I., Freeman, A.G., Walker, M.G., Blum, R.L., and Wiederhold, G. "Intelligent Summarization and Display of On-Line Medical Records" Accepted for publication in M.D. Computing.

29. DeZegher-Geets, I.M., Freeman, A.G., Walker, M.G., Blum, R.L., and Wiederhold, G.C.M.: "Computer-aided Summarization of a Time-oriented Medical Data Base"; in *Proceedings of the Third Annual Conference on Computerization of Medical Records.* Institute for Medical Record Economics, Chicago, April 1987. Also Stanford Knowledge Systems Laboratory (KSL) Report KSL-87-18.

30. DeZegher-Geets, I.: *Intelligent Summarization of a Time-Oriented Medical Data Base.* Master's thesis, Stanford University, 1987.

31. Downs, S., Walker, M.G., and Blum, R.L.: Automated Summarization of On-line Medical Records. In Proceedings of the Fifth World Congress on Medical Informatics (Medinfo), pages 800-804. Elsevier Science Publishers, 1986.

32. Downs, S.M.: *A Program for Automated Summarization of On-line Medical Records*, Master's thesis, Stanford University, 1986.

33. Kuhn, I., Wiederhold, G., Rodnick, J.E., Ramsey-Klee, D.M., Benett, S., Beck, D.D.: *Automated Ambulatory Medical Record Systems in the U.S.*, to be published by Springer-Verlag, 1983, in *Information Systems for Patient Care*, B. Blum (ed.), Section III, Chapter 14.

34. Shortliffe, E. H., Wiederhold, G., Fagan, L., and Perrault,L. : *An Introduction To Medical Computer Science.* Addison-Wesley, 1987. In preparation, several chapters completed.

35. Walker, M.G. "Expert Systems in Geologic Exploration: Can They Be Cost Effective?" Accepted for publication in Geobyte.

36. Walker, M.G. "Automated Recognition of Brain Anatomy and Pathology in MRI and CT: a Review". In preparation for IEEE Transactions on Medical Image Processing.

37. Walker, M.G., Blum, R.L., and Fagan, L.M. "Minimycin: A Miniature Rule-Based System" in "M.D.Computing Tutorials", C.J.McDonald (editor), Springer-Verlag, 1988. Reprinted from M.D.Computing, V.2,No.4, 1985.

38. Walker, M.G., and Blum, R.L. "An Introduction to LISP" in "M.D.Computing Tutorials", C.J.McDonald (editor), Springer-Verlag, 1988. Reprinted from M.D.Computing, V.2, No.1, 1985.

39. Walker, M.G. "How Feasible is Automated Discovery?" The Computer and Philosophy Newsletter, Vol.3, No.1, pp. 1-37, Feb 1988, Center for Design of Educational Computing, Carnegie Mellon University. Reprinted from IEEE Expert, Vol.2, No.1, pp. 70-82, Spring 1987.

40. Walker, M.G., Clauer, R.C., Samadani, R., Craven, J., and Frank, L. "Automated Analysis of Auroral Images"; Abstract. EOS, Vol. 68, No.

44, page 1436, November 1987. Poster session paper presented at American Geophysical Union Annual Meeting, San Francisco, December 1987.

41. Walker, M.G. "Expert Systems in Exploration: Can They Be Cost Effective?" American Association of Petroleum Geologists, Annual Convention, Los Angeles, California, June 1987.

42. Walker, M.G., and Blum, R.L.: *Towards Automated Discovery from Clinical Databases: the RADIX Project.* In Proceedings of the Fifth World Congress on Medical Informatics (Medinfo), pages 32-36. Elsevier Science Publishers, 1986.

43. Walker, M.G., Blum, R.L., and Fagan, L.M.: *Minimycin: A Miniature Rule-Based System.* M.D.Computing, Vol. 2, No. 4., 1985.

44. Walker, M.G., and Blum, R.L.: *An Introduction to LISP.* M.D. Computing, Vol. 2, No. 1., 1985.

45. Wiederhold, Gio: "Hospital Information Systems"; in Encyclopedia of Medical Devices and Instrumentation, vol.3, Webster,John G.(ed), Wiley 1988, pp.1517--1542

46. Wiederhold, Gio and Paul D. Clayton: "Processing Biological Data in Real Time"; M.D. Computing, Springer Verlag, Vol.2 No.6, November 1985, pages 16-25, republished in 'Images, Signals, and Devices', C.J. McDonald (editor), Springer Verlag, 1987, pp.107--116.

47. Wiederhold, G.: *File Organization for Database Design.* McGraw-Hill Book Company, 1987.

48. Wiederhold, Gio CM, Michael G. Walker, Robert L. Blum, Stephen M. Downs, and Isabelle deZegher-Geets: "Acquisition of Knowledge from Medical Records" abstract; in Namen, Daten Themen, Benutzer- gruppenseminar Medizin I, Systems 87 conference, Munich, FRG, Oct.1987, pp.213-214.

49. Wiederhold, Gio CM, Michael G. Walker, Waqar Hasan, Surajit Chaudhuri, Arun Swami, Sang K. Cha, XiaoLei Qian, Marianne Winslett, Linda DeMichiel, and Peter K. Rathmann: "KSYS: An Architecture for Integrating Databases and Knowledge Bases"; Computer Science Department, Stanford University, May 1987; in Amar Gupta and Stuart Madnick (editors) 'Technical Opinions Regarding Knowledge- Based Integrated Information Systems Engineering', MIT, 1987,

50. Wiederhold, Gio: "Knowledge versus Data"; Chapter 8 of 'On Knowledge Base Management Systems: Integrating Artificial Intelligence and Database Technologies' (Brodie, Mylopoulos, and Schmidt, eds.), Springer Verlag, June 1986, pages 77 to 82.

51. Wiederhold, Gio, Robert L. Blum, and Michael Walker: "An Integration of Knowledge and Data Representation"; Proc. of Islamorada Workshop, Feb.1985, Computer Corporation of America, Cambridge MA; Chapter 29 of 'On Knowledge Base Management Systems: Integrating Artificial Intelligence and Database Technologies' (Brodie, Mylopoulos, and Schmidt, eds.), Springer Verlag, June 1986, pages 431 to 444.

52. Wiederhold, G.: *Views, Objects, and Databases.* IEEE Computer 19(12):37-44, December, 1986.

53. Wiederhold, G.C.M., Walker, M. G., Blum, R.L., and Downs, S.M.: *Acquisition of Knowledge from Data.* In Ras, Z.W., and Zemankova, M. (editors), Proceedings of the International Symposium on Methodologies for Intelligent Systems, ACM, Oct.1986, Knoxville TN.

54. Missikoff, Michele and Gio Wiederhold: *Towards a Unified Approach for Expert and Database Systems.* In 'Expert Database Systems', Larry Kerschberg (editor), Benjamin/Cummings, 1986, pages 383-399; also in Proceedings of First Workshop on Expert Database Systems, Kiawah Island, South Carolina, Oct. 1984, vol.1, pp.186-206.

55. Wiederhold, Gio: *Knowledge Bases*; Future Generations Computer Systems, North-Holland, vol.1 no.4; April 1985, pp.223--235.

56. Wiederhold, Gio: *Disease Registers: Use of Databases to Generate New Medical Knowledge*; in K.Abt, W.Giere and B.Leiber: 'Krankendaten, Krankheitsregister, Datenschutz', vol.58, Medizinische Informatik und Statistik, Springer Verlag, 1985, pp.39-55.

57. Wiederhold, G.: *Networking of Data Information*, National Cancer Institute Workshop on the Role of Computers in Cancer Clinical Trials, National Institutes of Health, June 1983, pp.113-119.

58. Wiederhold, G.: *Database Design* (in the Computer Science Series); McGraw-Hill Book Company, New York, NY, May 1977, 678 pp. Second edition, Jan. 1983, 768 pp.

59. Wiederhold, G.: In D.A.B. Lindberg and P.L. Reichertz (Eds.), *Databases for Health Care*, Lecture Notes in Medical Informatics, Springer-Verlag, 1981.

60. Wiederhold, G.: "Database technology in health care"; J. Medical Systems 5(3):175-196, 1981.

*E. Funding Support Status*

1987-1988 Principal Investigator: Gio Wiederhold 5/20%
(6 months) DADAISM, Databases for Ada Information System Modules
(SRI//NRL/DoD Stars: $176,931)

1987-1991 Principal Investigator: Gio Wiederhold 10%
Support for Parallel Design in an Engineering Information
System (NSF DTMP 8619595 $153,766, $146,360 prop.,
$156,044 prop.)

1987-1988 Principal Investigator: Gio Wiederhold 5%
Validation of Knowledge from a Database (IBM KBS Menlo
Park, $50,051)

1986-1990 Principal Investigator: Gio Wiederhold 50% effort
Knowledge Base Management Structures (ONR/SPAWAR/ARPA,
N39-84-C-211, task 7: $1,756,410)

1987-1988 Principal Investigator: Gio Wiederhold 5%
Reasoning about R1ME
(Digital Equipment Corp: $131,337)

Requests in process:

1988 (6 mo) Principal Investigator: Gio Wiederhold 20% summer,
10% academic year
DADAISM: DBMS in and for ADA-supported Information
System Management (SRI/NRL/STARS/DoD: $336,719)

1987-1990 Principal Investigator: Gio Wiederhold 25% effort
FACS-Penguin: An Expert Workstation for Flow Cytometry
(NLM/NIH : $856,449 direct costs, approved at high priority)

1987-1990 Associate Investigator: Gio Wiederhold 10% effort
Integrating Knowledge and DBMS (SRI AI/ONR/SPAWAR/ARPA,
to be funded: $75,000 Stanford Subcontract)

1988-1990 Principal Investigator: Gio Wiederhold 10% effort
PARADATA: Databases on Parallel Computers (NSF/IRI/K&DSP,
approved, not yet funded, $499,385 requested)

1986-1989 Principal Investigator: Gio Wiederhold 5% effort
RADIX Project: Software for Automated Peer Review (NCHSR/
DHHS HS5632, approved, unlikely to be funded: $231,996 direct)

## II. INTERACTIONS WITH THE SUMEX-AIM RESOURCE

### *A. Collaborations*

SUMEX AIM provides the central communication node for our research. While individual experiments tend to take place on workstations, the availability of SUMEX to load and edit files, communicate with colleagues and peers is absolutely critical.

### *B. Interactions with Other SUMEX-AIM Projects*

During the current reporting year we have had frequent interaction with members of other SUMEX projects; for example, development of algorithms for transforming INTERNIST data to Bayes form, presentation of research results at Stanford Medical Information Science Colloquia, discussions of automated discovery and automated summarization, practical programming issues, and training of Medical Computer Science Students in the use of KEE, Lisp workstations, and so on. The SUMEX community is an invaluable resource for providing such interaction.

### *C. Critique of Resource Management*

The DECSystem 20 continues to provide acceptable performance, but it is frequently over loaded at peak hours. The archiving facility (SAFE) seems to be very loaded, so that we are forced to make decisions on retention of past material more frequently than optimal. As users of large databases this is one of the resources upon we must draw frequently.

The SUMEX resource management continues to be accessible and quite helpful. New networking links, for instance to the Genetics Department equipment, make new collaborations viable.

## III. RESEARCH PLANS

### *A. Project Goals and Plans*

The the RADIX Project is now inactive. We plan to keep the results and software produced accessible for educational and sharing objectives within the MIS program.

The long-range goal of the PENGUIN project is to integrate our experiment design advisory system with the set of programs developed in the Genetics department for FACS operation and control, thereby defining a comprehensive "FACS Workstation" that should considerably facilitate effective utilization of FACS technology. In the shorter term, we will:

1. complete the development and testing of the object interface. The interface should then be made a fully domain-independent module that could be applied in different contexts.
2. validate the structure of the database and make it available to the scientists in the Genetics department, for further refinement as well as for routine data storage and retrieval.
3. move gradually toward a distributed environment with Macintoshes as local workstations connected to a central VAX computer, acting as a database server.
4. develop the expert system module which will be coupled to the database through the object interface. The expert system module is aimed at assisting people in an experimental design task. This can actually be done by taking two different approaches, one being the actual automatic design of the experiment (i.e., a planning task), the other being the critique of a protocol proposed by the user. Although we recognize both directions as important, we will focus on the first approach because knowing how to properly plan new experiments is a necessary step prior to any critiquing process. Although the problem solving strategies of our experts are not yet fully determined, we plan to investigate the general concept of hierarchical planning.

### *B. Justification and Requirements for Use of SUMEX*

The PENGUIN project is dependent on new methods of communication among clinical and academic researchers.

### *C. Recommendations for Resource Development*

The movement towards work-stations is obviously the direction of the future. We hope that the communication capability, now supported via the central SUMEX facility, the DEC 2060, will continue to be supported. We continue to make constant use of this communication with our colleagues within and outside of Stanford, preparation of large documents, file and database handling, and program demonstrations.

# IV.B. National AIM Projects

The following group of projects is formally approved for access to the AIM aliquot of the SUMEX-AIM resource. Their access is based on review by the AIM Advisory Group and approval by the AIM Executive Committee.

In addition to the progress reports presented here, abstracts for each project and its individual users are submitted on a separate Scientific Subproject Form.

# IV.B.1. ATTENDING Project

## ATTENDING Project -- Expert Critiquing Systems

Perry L. Miller, M.D. Ph.D.
Department of Anesthesiology
Yale University School of Medicine
New Haven, CT 06510

## I. SUMMARY OF RESEARCH PROGRAM

### *A. Project rationale*

Our project is exploring the "critiquing" approach to bringing computer-based advice to the practicing physician.

Critiquing is a different approach to the design of artificial intelligence based expert systems. Most medical expert systems attempt to simulate a physician's decision-making process. As a result, they have the clinical effect of trying to tell a physician what to do: how to practice medicine. In contrast, a critiquing system first asks the physician how he contemplates approaching his patient's care, and then critiques that plan. In the critique, the system discusses any risks or benefits of the proposed approach, and of any other approaches which might be preferred. It is anticipated that the critiquing approach may be particularly well suited for domains, like medicine, where decisions involve a great deal of *subjective* judgment.

To date, several prototype critiquing systems have been developed in different medical domains, including:

1. ATTENDING, the first system to implement the critiquing approach, critiques anesthetic management.

2. HT-ATTENDING critiques the pharmacologic management of essential hypertension.

3. VQ-ATTENDING critiques aspects of ventilator management.

4. PHEO-ATTENDING critiques the laboratory and radiologic workup of a patient for a suspected pheochromocytoma.

In addition, a domain-independent system, ESSENTIAL-ATTENDING, has been developed to facilitate the implementation of critiquing systems in other domains.

### *C. Highlights of Research Progress*

Current projects include the following:

**HT-ATTENDING** -- The original prototype version of HT-ATTENDING has been converted to the ESSENTIAL-ATTENDING format, and updated to reflect current thinking in the field of hypertension management. A major priority is to subject this system to validation and clinical evaluation. In addition, we are currently exploring concretely how best to disseminate the system as a practical consultation tool.

**ICON: Critiquing Radiological Differential Diagnosis** -- Most existing diagnostic computer systems produce a ranked differential diagnosis as their output. In this process, the rich structure of the knowledge that went into developing the diagnoses may be lost to the user. ICON explores a different approach to diagnostic advice in the domain of radiology. To use ICON, a radiologist describes a set of findings seen on chest x-ray, together with a proposed diagnosis. ICON then produces a detailed analysis of *why* the observed findings serve to support or to rule out the diagnosis. It may also suggest further findings that might help refine the diagnosis, again explaining why the findings are important. In addition, we are currently exploring how images might be incorporated into ICON's knowledge base to enhance its consultative effectiveness. We see this project as an exploration of the question of how prose and images might best be indexed and organized for purposes of explanation in the context of a particular clinical case.

*D. Publications*

1. Miller, P.L. (Ed.): Selected Topics in Medical Artificial Intelligence. New York: Springer-Verlag (in press).

2. Rennels, G.D., Miller, P.L.: Artificial intelligence research in anesthesia and intensive care. Journal of Clinical Monitoring (in press).

3. Miller, P.L., Rennels, G.D.: Prose generation from expert systems: An applied computational linguistics approach. AI Magazine (in press).

4. Rennels, G.D., Shortliffe, E.H., Stockdale, F.E., Miller, P.L.: A computational model of reasoning from the clinical literature. Computer Methods and Programs in Biomedicine 24:139-149, 1987.

5. Rennels, G.D., Shortliffe, E.H., Stockdale, F.E., Miller, P.L.: A structured representation of the clinical literature and its use in a medical management advice system. Bulletin du Cancer 74:215-220, 1987.

6. Miller, P.L., Barwick, K.W., Morrow, J.S., Powsner, S.M., Riely, C.A.: Semantic relationships and medical bibliographic retrieval: A preliminary assessment. Computers and Biomedical Research 21:64-77, 1988.

7. Miller, P.L., Morrow, J.S., Powsner, S.M., Riely, C.A.: Semantically assisted medical bibliographic retrieval: An experimental computer system. Bulletin of the Medical Library Association 76:131-136, 1988.

8. Miller, P.L.: Exploring the critiquing approach: Clinical practice-based feedback by computer. Biomedical Measurement, Informatics and Control (in press).

9. Rennels, G.D., Shortliffe, E.H., Stockdale, F.E., Miller, P.L.: A computational model of reasoning from the clinical literature. The AI Magazine (accepted pending revision).

10. Miller, P.L., Fisher, P.R.: Causal models in medical artificial intelligence. Proceedings of the Eleventh Symposium on Computer Applications in Medical Care, Washington, D.C., November 1987, pp. 17-22.

11. Powsner, S.M., Barwick, K.W., Morrow, J.S., Riely, C.A., Miller, P.L.: Coding semantic relationships for medical bibliographic retrieval: A preliminary study. Proceedings of the Eleventh Symposium on Computer Applications in Medical Care, Washington, D.C., November 1987, pp. 108-112.

*E. Funding Support*

EXPERT COMPUTER SYSTEMS WHICH CRITIQUE PHYSICIAN PLANS
NIH Grant R01 LM04336
Principal Investigator: Perry L. Miller, M.D., Ph.D.
Annual Direct Costs: approximately $95,000
Period of Support: 3/1/88-2/28/91

> *This grant supports the exploration of the critiquing approach to bringing computer-based advice to the physician, focusing especially on refining and evaluating the HT-ATTENDING system which critiques hypertension management, and on developing tools for knowledge base maintenance and updating.*

SUPPORT OF THE UNIFIED MEDICAL LANGUAGE PROGRAM
NLM Contract N01 LM63524
Principal Investigator: Perry L. Miller, M.D., Ph.D.
Annual Direct Costs: approximately $100,000
Period of Support: 8/22/86-8/21/88

> *This research contract is part of the NLM Unified Medical Language (UML) program. We are defining a set of semantic relationships which could be used to augment the UML, to facilitate such functions as medical bibliographic retrieval.*

SUPPORT FOR MEDICAL INFORMATICS AND ARTIFICIAL INTELLIGENCE
Ira DeCamp Foundation
Co-Principal Investigators: Henry A. Swett, M.D.
Perry L. Miller, M.D., Ph.D.
Annual Costs: $75,000
Period of Support: 7/1/86-6/30/90

> *This grant supports activities of our Medical Informatics program.*

MEDICAL INFORMATICS RESEARCH TRAINING AT YALE
Principal Investigator: Perry L. Miller, M.D., Ph.D.
NLM Training Grant T15 LM07056
Period of Support: 7/1/87-6/30/92

> *This grant currently supports 3 postdoctoral trainees and three predoctoral trainees in Medical Informatics.*

*Pending Support*

BIOTECHNOLOGY COMPUTER RESEARCH AT YALE
Perry L. Miller, M.D., Ph.D. (PI)
RFA 88-LM-01

> *This grant proposes computer-related research in molecular biology.*

## II. INTERACTIONS WITH THE SUMEX-AIM RESOURCE

Until last year we were using the RUTGERS-AIM Resource. We used that facility to implement all of our early critiquing systems. This year we moved the HT-ATTENDING project. to the SUMEX-AIM facility. Very recently, we have moved HT-ATTENDING to a local VAX station, and we are currently migrating entirely to local workstations for our research. This past year, our main use of SUMEX-AIM has been the following:

1. We used SUMEX-AIM to refine and to demonstrate HT-ATTENDING.
2. We used SUMEX-AIM for communication access to the national AIM community.

We have found our use of the RUTGERS-AIM and SUMEX-AIM facilities to be extremely valuable. They provided us the resources needed to initiate our research and to continue several projects which are still active. They provided a natural vehicle to allow us to demonstrate the various systems easily, both in the United States and in Europe. Also, they enabled us to collaborate very closely with Dr. Glenn Rennels in his Stanford Medical Information Science thesis project on the Roundsman system. Via SUMEX-AIM and RUTGERS-AIM, Dr. Rennels and Dr. Miller maintained very close contact, typically with multiple messages each week, and sometimes within a single day.

### III. FUTURE PLANS

We plan to continue our critiquing research as outlined above. As discussed previously, we have now moved all our research onto local workstations, and will continue the research using these resources.

# IV.B.2. INTERNIST-I/QMR Project

## INTERNIST-I/QMR (Quick Medical Reference)

Jack. D. Myers, M.D.
University Professor Emeritus (Medicine)

Randolph A. Miller, M.D.
Associate Professor of Medicine
Chief, Section of Medical Informatics

University of Pittsburgh
1291 Scaife Hall
Pittsburgh, Pa., 15261

## I. SUMMARY OF RESEARCH PROGRAM

### *A. Project rationale*

The principal objective of this project is the development of a high-level computer diagnostic program in the broad field of internal medicine as an aid in the solution of complex and complicated diagnostic problems. To be effective, the program must be capable of multiple diagnoses (related or independent) in a given patient.

A major achievement of this research undertaking has been the design of a program called INTERNIST-I, along with an extensive medical knowledge base. This program has been used over the past decade to analyze many hundreds of difficult diagnostic problems in the field of internal medicine. These problem cases have included cases published in medical journals (particularly Case Records of the Massachusetts General Hospital, in the New England Journal of Medicine), CPCs, and unusual problems of patients in our Medical Center. In most instances, but by no means all, INTERNIST-I has performed at the level of the skilled internist, but the experience has highlighted several areas for improvement.

### *B. Medical Relevance and Collaboration*

The program inherently has direct and substantial medical relevance.

The development of the QUICK MEDICAL REFERENCE (QMR) under the leadership of Dr. Randolph A. Miller has allowed us to distribute the INTERNIST-I knowledge base in a modified format to over twenty other academic medical institutions. The knowledge base can thereby be used as an "electronic textbook" in medical education at all levels -- by medical students, residents and fellows, and faculty and staff physicians. This distribution is continuing to expand.

The INTERNIST-I program has been used in recent years to develop patient management problems for the American College of Physician's Medical Knowledge Self-assessment Program.

## *C. Highlights of Research Progress*

### C.1 Accomplishments this past year

For the record, it should be noted that grant support for the QMR project has come solely from the CAMDAT Foundation of Farmington, Conn., from the Department of Medicine of the University of Pittsburgh, and from Dr. Miller's NLM RCDA grant and NLM RO1 grant.

In the past year, the University of Pittsburgh was named recipient of a National Library of Medicine Medical Informatics Training Grant Award.

The group of us (Myers, Miller and Masarie) together with assigned residents in internal medicine and fellows in medical informatics are continuing to expand the knowledge base and to incorporate the diagnostic consultative program into QMR. The computer program for the interrogative part of the diagnostic program is the main remaining task. An editor for the QMR knowledge base, as modified from the INTERNIST-I knowledge base, has been written from scratch in Turbo Pascal by Dr. Masarie. The entire QMR program can be accommodated in, maintained (particularly edited) and operated on individual IBM PC-AT computers.

Our group has incorporated into the QMR diagnostic consultant program modifications and embellishments of the INTERNIST-I knowledge base, and will continue to do so over the next year by adding "facets" of diseases or syndromes. This addition and modification is expected to improve the performance of the diagnostic consultant program.

The medical knowledge base has continued to grow both in the incorporation of new diseases and the modification of diseases already profiled so as to include recent advances in medical knowledge. Several dozen new diseases have been profiled during the past year. The current number of diseases in the QMR knowledge base is 589, and 4237 possible patient findings are included.

### C.2 Research in progress

There are four major components to the continuation of this research project:

1. The enlargement, continued updating, refinement and testing of the extensive medical knowledge base required for the operation of INTERNIST-I and the QMR modification.

2. Institution of field trials of QMR on the clinical services in internal medicine at the Health Center of the University of Pittsburgh. This has been accomplished in a limited fashion, which began in 1987; a "computer-based diagnostic consultation service" has been made available to attending physicians and house staff on the medical services of our two main teaching hospitals. Institutional Review Board (IRB) approval was granted to the service before it was initiated.

3. Expansion of the clinical field trials to other university health centers which have expressed interest in working with the system.

4. Adaptation of the diagnostic program and data base of INTERNIST-I and the QMR modification to subserve educational purposes and the evaluation of clinical performance and competence.

Current activity is devoted mainly to the first two of these, namely, the continued development of the medical knowledge base, and the implementation of the improved diagnostic consulting program, and preliminary evaluation of the diagnostic consultation service.

*D. List of relevant publications*

1. Myers JD. The Background of INTERNIST-I and QMR. In: Proceedings of the History of Medical Informatics Conference. National Library of Medicine. pp. 195-197, November 1987.
2. Masarie, F.E., Miller, R.A. Medical Subject Headings and Medical Terminology: An analysis of terminology used in hospital charts. Bulletin of the Medical Library Association, 1987; 75:89-94.
3. Masarie FE, Miller RA. INTERNIST-I to Quick Medical Reference (QMR): Transition from a mainframe to a microcomputer. Proceedings Ninth Annual IEEE/Engineering in Biology and Biology Society. IEEE Press. pp. 1521- 1522, November 1987.
4. Parker, RC, Miller RA. Using causal knowledge to create simulated patient cases: The CPCS project as an extension of INTERNIST-I. Proceedings of the Eleventh Annual Symposium on Computer Applications in Medical Care. pp. 473-480, November 1987.
5. Bankowitz RA, Blumenfeld BH, Miller RA, et al. User variability in abstracting and entering printed case histories with Quick Medical Reference (QMR). Proceedings of the Eleventh Annual Symposium on Computer Applications in Medical Care. pp. 68-73, November 1987.
6. Masarie FE, Miller RA. Quick Medical Reference (QMR): An information management tool for clinical diagnosis. IN: Critical Reviews in Medical Informatics. Boca Raton, Florida, CRC Press, 1988.
7. Parker RC, Miller RA. Using causal knowledge to created simulated patient cases: The CPCS project as an extension of INTERNIST-I. IN: Miller PL (ed) Topics in Medical Artificial Intelligence. Computers and Medicine Series, Springer-Verlag, New York, 1988.
8. Challinor SM, McNeil MA, Bankowitz RA, et al. Evaluation of a Computer-Assisted General Medicine Diagnostic Consultation Service. Abstract presented at the 11th Annual SGIM Meeting, Washington, D.C., April 27-29, 1988.
9. Miller RA. From Automated Medical Records to Expert System Knowledge Bases: Common Problems in Representing and Processing Patient Data. Topics in Health Record Management. 75(3):23-36, 1987
10. Miller RA. Computer-based Diagnostic Decision-making. Medical Care. 25(12):S148-S152, 1987.

*E. Funding support*

1. Diagnostic-Internist: A Computerized Medical Consultant
   Randolph A. Miller, M.D.

Associate Professor of Medicine
Chief, Section of Medical Informatics
University of Pittsburgh Department of Medicine
National Library of Medicine - Development Award Research Career
National Institutes of Health

5 KO4 LM00084-03
09/30/85 - 09/29/86 - $55,296
09/30/86 - 09/29/87 - $55,296
09/30/87 - 09/29/88 - $54,648
Support recommended for 2 additional years ending 09/29/90,
The Amounts to be determined annually.

2. Developing INTERNIST-I Knowledge Base into a Resource
Randolph A. Miller, M.D.
Associate Professor of Medicine
Chief, Section of Medical Informatics
University of Pittsburgh Department of Medicine
National Library of Medicine
National Institutes of Health

1 RO1 LM04622-01
09/30/87 through 09/29/90
09/30/87 - 09/29/88 - $71,892
09/30/88 - 09/29/89 - $112,938
09/30/89 - 09/29/90 - $112,580

3. Pittsburgh Medical Information Sciences Training Program
Randolph A. Miller, M.D.
Associate Professor of Medicine
Chief, Section of Medical Informatics
University of Pittsburgh Department of Medicine
National Library of Medicine
National Institute of Health
07/01/87 through 06/30/92
07/01/87 - 06/30/88 - $153,454
07/01/88 - 06/30/89 - $199,038
07/01/89 - 06/30/90 - $217,572
07/01/90 - 06/30/91 - $278,092
07/01/91 - 06/30/92 - $276,596

## II. INTERACTIONS WITH THE SUMEX-AIM RESOURCE

### *A,B. Medical Collaborations and Program Dissemination Via SUMEX*

INTERNIST-I and QMR remain in a stage of research and particularly development. As noted above, we are continuing to develop better computer programs to operate the diagnostic system, and the knowledge base cannot be used very effectively for collaborative purposes until it has reached a critical stage of completion. These factors have stifled collaboration via SUMEX up to this point and will continue to do so for the next year or two. In the meanwhile, through the SUMEX community there continues to be an exchange of information and states of progress. Such interactions particularly take place at the annual AIM Workshop.

### *C. Critique of Resource Management*

SUMEX has been an excellent resource for the development of INTERNIST-I. Our large program is handled efficiently, effectively and accurately. The staff at SUMEX have been uniformly supportive, cooperative, and innovative in connection with our project's needs.

## III. RESEARCH PLANS

### *A. Project Goals and Plans*

Continued effort to complete the medical knowledge base in internal medicine will be pursued including the incorporation of newly described diseases and new or altered medical information on "old" diseases. The latter two activities have proven to be more formidable than originally conceived.

### *B. Justification and Requirements for Continued SUMEX Use*

Our use of SUMEX has declined with the adaptation of our programs to the IBM PC-AT. Nevertheless, the excellent facilities of SUMEX are expected to be used for certain developmental work. It is intended for the present to keep INTERNIST-1 at SUMEX for comparative use as QMR is developed here. We will not need the DEC 2060 beyond its anticipated phase-out in early 1989, but will require access to its replacement for mailing purposes and to maintain contact with the national medical informatics community.

### *C. Needs and Plans for Other Computing Resources Beyond SUMEX-AIM*

Our predictable needs in this area will be met by our recently acquired personal work stations.

# IV.B.3. MENTOR Project

**MENTOR Project -- Medical Evaluation of Therapeutic Orders**

**Stuart M. Speedie, Ph.D.**
**School of Pharmacy**
**University of Maryland**

**Terrence F. Blaschke, M.D.**
**Department of Medicine**
**Division of Clinical Pharmacology**
**Stanford University**

## I. SUMMARY OF RESEARCH PROGRAM

### *A. Project Rationale*

The goal of the MENTOR (Medical EvaluatioN of Therapeutic ORders) project is to design and develop an expert system for monitoring drug therapy for hospitalized patients that will provide appropriate advice to physicians concerning the existence and management of adverse drug reactions. The computer as a record-keeping device is becoming increasingly common in hospital-based health care, but much of its potential remains unrealized. Furthermore, this information is provided to the physician in the form of raw data which is often difficult to interpret. The wealth of raw data may effectively hide important information about the patient from the physician. This is particularly true with respect to adverse reactions to drugs which can only be detected by simultaneous examinations of several different types of data including drug data, laboratory tests and clinical signs.

In order to detect and appropriately manage adverse drug reactions, sophisticated medical knowledge and problem solving is required. Expert systems offer the possibility of embedding this expertise in a computer system. Such a system could automatically gather the appropriate information from existing record-keeping systems and continually monitor for the occurrence of adverse drug reactions. Based on a knowledge base of relevant data, it could analyze incoming data and inform physicians when adverse reactions are likely to occur or when they have occurred. The MENTOR project is an attempt to explore the problems associated with the development and implementation of such a system and to implement a prototype of a drug monitoring system in a hospital setting.

### *B. Medical Relevance and Collaboration*

A number of independent studies have confirmed that the incidence of adverse reactions to drugs in hospitalized patients is significant and that they are for the most part preventable. Moreover, such statistics do not include instances of suboptimal drug therapy which may result in increased costs, extended length-of-stay, or ineffective therapy. Data in these areas are sparse, though medical care evaluations carried out as part of hospital quality assurance programs suggest that suboptimal therapy is common.

Other computer systems have been developed to influence physician decision making by monitoring patient data and providing feedback. However, most of these systems suffer from a significant structural shortcoming. This shortcoming involves the evaluation rules that are used to generate feedback. In all cases, these criteria consist of discrete, independent rules, yet medical decision making is a complex process in which many factors are interrelated. Thus, attempting to represent medical decision-making as a discrete set of independent rules, no matter how complex, is a task that can, at best, result in a first-order approximation of the process. This places an inherent limitation on the quality of feedback that can be provided. As a consequence it is extremely difficult to develop feedback that explicitly takes into account all information available on the patient. One might speculate that the lack of widespread acceptance of such systems may be due to the fact that their recommendations are often rejected by physicians. These systems must be made more valid if they are to enjoy widespread acceptance among physicians.

The MENTOR system is designed to address the significant problem of adverse drug reactions by means of a computer-based monitoring and feedback system to influence physician decision-making. It employs principles of artificial intelligence to create a more valid system for evaluating therapeutic decision-making.

The work in the MENTOR project is a collaboration between Dr. Blaschke at Stanford University, Dr. Speedie at the University of Maryland, and Dr. Charles Friedman at the University of North Carolina. Dr. Speedie provides the expertise in the area of artificial intelligence programming. Dr. Blaschke provides the medical expertise. Dr. Friedman contributes expertise in the area of physician feedback design and system impact evaluation. The blend of previous experience, medical knowledge, computer science knowledge and evaluation design expertise they represent is vital to the successful completion of the activities in the MENTOR project.

### *C. Highlights of Research Progress*

The MENTOR project was initiated in December, 1983. The project has been funded by the National Center for Health Services Research since January 1, 1985. Initial effort focused on exploration of the problem of designing the MENTOR system. As of June 1, 1988, a working prototype system has been developed and is undergoing evaluation. The prototype consists of a Patient Data Base, an Inference Engine, an Advisory Module and a Medical Knowledge Base. The Medical Knowledge Base currently contains information related to aminoglycoside therapy, digoxin therapy, potassium supplementation, surgical prophylaxis, and microbiology lab reports. The system is currently implemented on a Xerox 1186 AI Workstation. Another version of the Patient Data Base has been developed for a VAX 750 that is connected via an asynchronous line to the 1186 running the inference engine. The project has received additional funding from the National Center for Health Services Research to install and evaluate the MENTOR system in a Veterans Administration Hospital. This effort began in June of 1988 and will continue for two additional years. The VA system will reside on an 1186 and a VAX Station II connected directly to the VA's Ethernet LAN, and accessing hospital data through the FILEMAN software.

### *E. Funding Support*

Title: MENTOR: Monitoring Drug Therapy for Hospitalized Patients

Principal Investigators:

Terrence F. Blaschke, M.D.
Division of Clinical Pharmacology

Department of Medicine
Stanford University

Stuart M. Speedie, Ph.D.
School of Pharmacy
University of Maryland

Funding Agency: National Center for Health Services Research

Grant Identification Number: 1 R18 HS05263

Total Award: January 1, 1985 - May 31, 1990 $1,056,201 Total
Direct Costs

Current Period: June 1, 1988 - May 31, 1989 $396,569 Total
Direct Costs

## II. INTERACTIONS WITH THE SUMEX-AIM RESOURCE

### *A. Medical Collaborations and Program Dissemination via SUMEX*

This project represents a collaboration between faculty at Stanford University Medical Center, the University of Maryland School of Pharmacy, and the University of North Carolina in exploring computer-based monitoring of drug therapy. SUMEX, through its communications capabilities, facilitates this collaboration of geographically separated project participants by providing electronic mail and file exchange between sites.

### *B. Sharing and Interactions with Other SUMEX-AIM Projects*

Interactions with other SUMEX-AIM projects has been on an informal basis. Personal contacts have been made with individuals working on the ONCOCIN project concerning system development issues. Dr. Perry Miller has also been of assistance by providing software for advisory generation. Given the geographic separation of the investigators, the ability to exchange mail and programs via the SUMEX system as well as communicate with other SUMEX-AIM projects is vital to the success of the project.

### *C. Critique of Resource Management*

To date, the resources of SUMEX have been fully adequate for the needs of this project. The staff have been most helpful with any problems we have had and we are quite satisfied with the current resource management.

## III. RESEARCH PLANS

### *A. Project Goals and Plans*

The MENTOR project has the following goals:

1. Implement a prototype computer system to continuously monitor patient drug therapy in a hospital setting. This will be an expert system that will use a modular, frame-oriented form of medical knowledge, a separate inference engine for applying the knowledge to specific situations, and automated collection of data from hospital information systems to produce therapeutic advisories.

2. Select a small number of important and frequently occurring medical settings (e.g., combination therapy with cardiac glycosides and diuretics) that can lead to therapeutic misadventures, construct a comprehensive medical knowledge base necessary to detect these situations using the information typically found in a computerized hospital information system and generate timely advisories intended to alter behavior and avoid preventable drug reactions.

3. Design and begin to implement an evaluation of the impact of the prototype MENTOR system on physicians' therapeutic decision-making as well as on outcome measures related to patient health and costs of care.

1988 will be spent on continued prototype development in six content areas, refinement of the inference mechanisms, and installation of the system at the Palo Alto Veterans Administration Hospital.

### *B. Justification and Requirements for Continued SUMEX Use*

This project needs continued use of the SUMEX facilities for one primary reason. Access to SUMEX is necessary to support the collaborative efforts of geographically separated development teams at Stanford and the University of Maryland.

Furthermore, the MENTOR project is predicated on the access to the SUMEX resource free of charge over the next two years. Given the current restrictions on funding, the scope of the project would have to be greatly reduced if there were charges for use of SUMEX.

### *C. Needs and Plans for Other Computing Resources Beyond SUMEX-AIM*

A major long-range goal of the MENTOR project is to implement this system on a independent hardware system of suitable architecture. It is recognized that the full monitoring system will require a large patient data base as well as a sizeable medical knowledge base and must operate on a close to real-time basis. Ultimately, the SUMEX facilities will not be suitable for these applications. Thus, we have transported the prototype system to a dedicated hardware system that can fully support the the planned system and which can be integrated into a Hospital Information System. For this purpose a VAX 750 and three Xerox 1186 workstations have been acquired and our development efforts have been transferred to them.

### *D. Recommendations for Future Community and Resource Development*

In the time we have been associated with SUMEX, we have been generally pleased with the facilities and services. However, it is clearly evident that the users' almost insatiable demands for CPU cycles and disk space cannot be met by a single central machine. The best strategy would appear to be one of emphasizing powerful workstations or relatively small, multi-user machines linked together in a nationwide network with SUMEX serving as the its central hub. This would give the individual users much more control over the resources available for their needs, yet at the same time allow for the communications among users that have been one of SUMEX's strong points.

For such a network to be successful, further work needs to be done in improving the network capabilities of SUMEX to encourage users at sites other than Stanford. Further work is also needed in the area of personal workstations to link them to such a network. Given the successful completion of this work, it would be reasonable to consider the phase-out of the central SUMEX machine years and its replacement by an efficient, high-speed communications server.

## IV.C. Pilot Stanford Projects

Following are descriptions of the informal pilot projects currently using the Stanford portion of the SUMEX-AIM resource, pending funding, full review, and authorization.

In addition to the progress reports presented here, abstracts for each project are submitted on a separate Scientific Subproject Form.

# IV.C.1. REFEREE Project

## REFEREE Project

**Principal Investigator: Bruce G. Buchanan, Ph.D.**
**Computer Science Department**
**Stanford University**

**Co-Principal Investigator: Byron W. Brown, Ph.D.**
**Department of Medicine**
**Stanford University**

**Associate Investigator: Daniel E. Feldman, Ph.D., M.D.**
**Department of Medicine**
**Stanford University**

## I. SUMMARY OF RESEARCH PROGRAM

### *A. Project Rationale*

The goals of this project are related both to medical science and artificial intelligence: (a) use AI methods to allow the informed but non-expert reader of the medical literature to evaluate a randomized clinical trial, and (b) use the interpretation of the medical literature as a test problem for studies of knowledge acquisition and fusion of information from disparate sources. REFEREE and REVIEWER, a planned extension, will be used to evaluate the medical literature of clinical trials to determine the quality of a clinical trial, make judgments on the efficacy of the treatment proposed, and synthesize rules of clinical practice. The research is an initial step toward a more general goal - building computer systems to help the clinician and medical scientist read the medical literature more critically and more rapidly for use in making clinical decisions.

### *B. Medical Relevance*

The explosive growth of the medical literature has created a severe information gap for the busy clinician. Most physicians can afford neither the time required to study all the pertinent journal articles in their field, nor the risk of ignoring potentially significant discoveries. The majority of clinicians, in fact, have little sophistication in epidemiology and statistics; they must nonetheless base their pragmatic decisions on a combination of clinical experience and published literature. The clinician's computerized assistant must ferret out useful maxims of clinical practice from the medical literature, pass judgment on the quality of medical reports, evaluate the efficacy of proposed treatments, and adjudicate the interpretation of conflicting and even contradictory studies.

### *C. Highlights of Progress*

REFEREE presently encodes the methodological knowledge of a highly regarded biostatistician at Stanford (Dr. Bill Brown). The system allows the informed but non-expert reader of the medical literature to evaluate the credibility of a randomized clinical trial.

In the future, REFEREE and its extensions will alleviate the knowledge-acquisition bottleneck for an automated medical decision-maker: the program will evaluate the quality of a clinical trial, judge the efficacy of the treatment proposed therein, and synthesize rules of clinical practice. For the present, however, the fusion of knowledge from disparate sources remains a problem in pure AI. The efforts of the REFEREE team have instead focused their efforts on the refinement and deepening of REFEREE's biostatistical knowledge by applying effective knowledge acquisition and knowledge engineering techniques. Dr. Diana Forsythe and Dr. Harold Lehmann are developing and using interview methods to acquire this knowledge from Dr. Brown, and R. Martin Chavez is implementing this in the prototype REFEREE expert system.

The REFEREE prototype is a consultant that evaluates the design and reporting of a single conclusion from randomized control trial for its credibility. It contains, in preliminary form, Professor Brown's expert knowledge of biostatistics. REFEREE evaluates each statistical procedure described by the authors of the paper. The automated consultant then determines the most appropriate method for the problem at hand, based on the design of the trial and the hypotheses to be tested. REFEREE checks critical assumptions, looks for possible statistical abuses, and verifies adjustments.

*The Knowledge Base:* Randomized controlled trials are used to test hypotheses regarding the effectiveness of various kinds of medical interventions. Dr. Brown classifies studies on the basis of three major attributes: the type of intervention tested (e.g. drug, surgery, health process change, etc.); the type of endpoint against which that intervention was tested (e.g. mortality, objective morbidity, subjective morbidity, etc.); and the type of conclusion drawn by the investigator/author on the basis of the research (e.g. that different treatments do or do not produce different outcomes, that a particular treatment is or is not cost-effective, etc.). Following this classificatory scheme, we decided to begin by producing a prototype REFEREE system that would help the reader to evaluate a single published conclusion concerning the effect of a given drug treatment on mortality.

*Knowledge Acquisition:* Having defined the scope of the initial knowledge base, we turned to the problem of collecting the information from Dr. Brown for inclusion in the system, i.e. knowledge acquisition. This task generally involves a relatively long-term process of face-to-face information gathering during sessions between the expert and one or more knowledge engineers. Dr. Diana Forsythe has noted a parallel between the communicative and analytical tasks involved in knowledge acquisition and those undertaken in ethnographic research. For this reason, we included an anthropologist in the research team and make use of ethnographic techniques in order to maximize the efficiency and quality of the data collection process.

Dr. Lehmann and Dr. Forsythe have carried out several months of systematic interviews with Dr. Brown in order to begin the process of constructing and refining the knowledge base for the current REFEREE prototype. We have combined a case-based approach that allows us actively to observe Dr. Brown as he reads papers, with semi-directed interviewing oriented toward understanding his terminology and category system. We find that these techniques work very well: Dr. Brown's interest in the knowledge acquisition process has been sustained, and indeed has increased over time as the system based on his expertise has evolved. He is clearly comfortable with this approach, and notes that it has actually afforded him additional insight into the way he interprets the literature.

Over the past year we have altered our knowledge representation from that of rules to that of an *influence diagram*. This is an acyclic directed graph of propositions or variables connected by links, where the absence of a link indicates conditional independence of the two variables. This formalism has been used in decision analysis to enable experts to convey their knowledge of a domain, and, more recently, has been used in AI to represent that knowledge in expert systems. This shift in formalism significantly altered the knowledge acquisition process and the implementation of that knowledge in our program.

Based on information from our expert, we have taken *credibility* as the goal parameter of the present system. This goal is defined operationally by Dr. Brown as "my odds that the conclusion of the paper would be replicated in an experiment based on the methods reported in the paper but without any of the flaws". In assessing *credibility*, for instance, Dr. Brown considers the blindedness of the randomization, the blindedness of the execution, the equivalence of the two groups at baseline, the equivalence in treatment of the two groups, the completeness of results reporting, and the propriety of the statistical analysis. We recognize that these variables are not all conditionally independent on *credibility*; work is in progress to assess as accurately as possible just what the conditional relationships are. Our use of influence diagrams has numerous advantages: the approach is acceptable to Dr. Brown, it is flexible, it can represent several aspects of the structure of the knowledge used by the expert, and the resultant data can be entered easily into the computer.

*Inference in REFEREE:* REFEREE was originally built within EMYCIN, a backward-chaining rule-based AI environment developed from MYCIN at Stanford. This environment is ideally suited for ordered collection of evidence and a diagnosis of the goal state at the end of that process. The state of belief or knowledge in parameters not directly between the evidence and the goal state is irrelevant. Our present system focuses on maintaining consistency over the entire knowledge base as new evidence is incorporated into the system. The constraints implied by the new data and Dr. Brown's prior knowledge are propagated throughout the system by Judea Pearl's message-passing algorithm for belief networks. During the consultation with the program, questions are chosen by the user and answered at his or her discretion, and the state of belief in any parameter can be requested at any time. The odds of replicating the study, then, can be viewed at any point during evidence collection.

*The User Interface:* REFEREE was initially run entirely on the SUMEX resource. Mr. Chavez reimplemented the program on a stand-alone workstation, the Xerox 1186 in the KEE commercial expert system shell. The availability of bit-mapped screens made us more sensitive to issues of the user interface, but the shell could not deal easily with the uncertainty inherent in our domain. Mr. Chavez then ported the system to a Texas Instrument Explorer work-station, for which he designed an entirely new knowledge engineering shell which integrated EMYCIN and influence diagrams. It was apparent, however, that to accommodate the multiple interface needs of our potential user community, we needed a graphics environment that would allow frequent changes and customization. Thus, we turned to a final environment custom-made for influence-diagram-based expert systems. The KNET system, also developed by Mr. Chavez, separates the inferencing capabilities and graphical manipulation of the knowledge base into MPW Object Pascal from the textual part of the knowledge base and the the evidence collection in HyperCard. This system runs on the Macintosh II with 4 MB of RAM.

The program code is now entirely independent of the knowledge required for reading papers. REFEREE has a new interface that is intuitive and consistent. There is an innovative consultation mode in which questions are presented in free-format menus. The dialogues are mixed-initiative and of mixed levels, allowing the user such options as requesting more detailed questions or cutting off apparently fruitless lines of questioning. With the new REFEREE prototype, the user interacts with the machine using a mouse-pointing device Finally, the screen enables the user to orient himself at all times, obviating the need for special commands to help the user "navigate" through the knowledge base. Our expert recently provided the best indication of the usability of this new system. After only a brief introduction to the new machine and interface, he was able - for the first time - to run an entire consultation by himself.

*Current Status:* At this point, REFEREE is a prototype that enables the clinician to read clinical trials more critically. A number of computational issues remain, such as the optimal representation of Dr. Brown's knowledge in our current formalism. Furthermore, REFEREE represents only the first step in a larger research plan, the automation of knowledge acquisition (see section on Research Plans, below). Current work in the restricted domain of clinical trials will, we hope, illustrate general principles in the design of decision makers that gather expertise from written text and multiple knowledge sources.

### *D. Relevant Publications*

1. Haggerty, J.: *REFEREE and RULECRITIC: Two prototypes for assessing the quality of a medical paper.* REPORT KSL-84-49. Master's Thesis, Stanford University, May 1984.
2. *Chavez, R. Martin and Cooper, G. F.: *KNET: Integration Hypermedia and Normative Bayesian Modeling.* REPORT KSL. Stanford University, March 1988.
3. *Lehmann, H. *Knowledge Acquisition for Probabilistic Expert Systems.* Submitted to Symposium on Computer Applications in Medical Care, 1988.

### *E. Funding Support*

REFEREE currently receives only a small amount of funding. Most of the research is performed in time contributed by the researchers to this project.

Title: Knowledge-Based Systems Research
PI: Edward A. Feigenbaum
Agency: Defense Advanced Projects Research Agency
Grant identification number: N00039-86-0033
Total award period and amount: 10/1/85 - 9/30/88 $4,130,230
(direct and indirect)
Current award period and amount: 10/1/87 - 9/30/88 $1,467,300
(direct and indirect)
REFEREE component is $27,706, or 1.9 % of grant total.

## II. INTERACTIONS WITH THE SUMEX-AIM RESOURCE

### *A. Medical Collaborations*

Dr. Brown and Dr. Feldman of the Stanford University School of Medicine are actively involved in the REFEREE project and are the primary domain experts and critics for this project.

*C. Critique of Resource Management*

The SUMEX computer resource and Lisp workstations have been very important for the work to date, and the SUMEX staff has continued to be very cooperative with the REFEREE project.

## III. RESEARCH PLANS

*A. Goals & Plans*

The overall objective of the REFEREE project is to use recent Artificial Intelligence techniques to build a system that helps the informed but statistically non-expert reader to evaluate critically the medical literature on randomized controlled trials (RCT's). This system will contain and be able to apply dynamically the detailed specialized knowledge of Dr. Byron W. Brown, a biostatistician expert in the design and evaluation of randomized controlled trials. We have divided our overall objective into two goals:

- Goal 1 is the construction of an expert system to help readers (e.g. medical students, medical researchers, clinicians, journal editors, or editorial assistants) assess the credibility of a *single* conclusion drawn from a *single* journal report of .a randomized controlled trial. We have already made substantial progress toward this goal with the development of the prototype REFEREE system.
- Goal 2 is the expansion of REFEREE to an expert system that can be used by a similar range of readers to facilitate the evaluation of *multiple* reports based on randomized controlled trials. This expanded system, to be known as the REVIEWER, will thus perform meta-analysis.

The task of extending and refining the prototype REFEREE system in order to achieve these goals can be characterized in terms of three dimensions:

- Making the system more accessible to a variety of people by improving the user interface, validating the system's performance with different types of users, and providing an explanatory capability
- Expanding the knowledge base by continuing the knowledge acquisition process to cover additional types of RCT's
- Improving the inference engine to ensure consistency of the knowledge base and to focus the consultation process on questions relevant to the situation and the individual user.

The specific steps that are planned for the enhancement of the REFEREE system include the following:

- Critique individual clinical trials according to the methodological quality of the trial;
- Measure the efficacy of treatment as demonstrated in a randomized control trial;
- Compare and contrast the credibility and efficacy of treatment reported by multiple journal articles; and

- Combine the *qualitative* techniques of heuristic reasoning and the *quantitative* methods of statistical meta-analysis to extract a consensus opinion from multiple knowledge sources.

In addition, plans for Goal 2, the REVIEWER system to analyze multiple RCT's and form a consensus judgment, include:

- Complete a review of the available literature on meta-analysis and augment the REFEREE prototype to produce estimators for meta-analysis and incorporate expert knowledge on the appropriateness of these methods.
- Add explicit and heuristic knowledge needed for the calculation of robust, non-parametric estimators of effect size.
- Construct a prototype of a system that builds categorical models in the domain of meta-analysis, to perform autonomous investigations in the domain of statistical model-building. The REVIEWER will utilize expert knowledge in biostatistics to guide its search for meaningful models.
- Build a prototype of a system that can explore the domain of regression models for multiple RCT's that will use expert knowledge in its selection of predictor variables.
- Package the REVIEWER in a form suitable for use by physicians and their assistants.
- Verify the expertise of the REVIEWER system on a suite of papers drawn from clinical trials, similar to the validation of REFEREE above.

### *B. Justification for continued SUMEX use*

The local area network maintained by the SUMEX staff is essential to the effective development and use of the REFEREE system on Lisp workstations. The availability of the Xerox workstations makes possible the evaluation of prototypes in that environment, and also facilitates the development of good user interfaces. The connections through the 2060 to local and national computer networks such as ARPANET are important for sharing ideas and results with other medical researchers.

### *C. Need for other computing resources*

We are exploring the implementation of REFEREE on personal computers such as the Macintosh II and other high performance machines. We anticipate the need for at least two of these machines for transporting our system and developing new modes of interaction with both naive and experienced users.

## IV.D. Pilot AIM Projects

Following is a description of the informal pilot projects currently using the AIM portion of the SUMEX-AIM resource, pending funding, full review, and authorization.

In addition to the progress report presented here, an abstract is submitted on a separate Scientific Subproject Form.

# IV.D.1. Dynamic Systems Project

## Decision Support for Time-Varying Clinical Problems

Lawrence E. Widman, M.D., Ph.D.
Division of Cardiology
University of Texas Health Science Center
San Antonio, Texas 78284

## I. SUMMARY OF RESEARCH PROGRAM

### *A. PROJECT RATIONALE*

Time-varying systems, which include many areas of medicine, science, economics, and business, can be described mathematically by differential equations. They are distinct from the pattern-matching and logic-based domains dealt with so successfully by existing expert system methods, because they can include feedback relationships. It is generally felt that they are best approached by enhancement of existing methods for deep model-based reasoning.

The goal of this project is to develop AI methods for capturing and using knowledge about time-varying systems. The strategy is to address general problems in model-based knowledge representation and reasoning. The intermediate objective is to develop methods which are powerful enough to work in selected realistic situations yet are general enough to be transportable to other, unrelated knowledge domains.

Our approach is to work on well-defined yet complex and interesting problems in the medical domain. We have, therefore, selected the human cardiovascular system as our prototype of a time-varying system, and are developing methods for representing and reasoning about its mechanical and electrical activities in the normal and diseased states.

### A.1 Technical Goals

This project presently has two distinct tracks: hemodynamic modeling and cardiac arrhythmia interpretation.

1. Hemodynamic Modeling

The goals of this subproject are to develop:

(a) a knowledge-representation method using symbolic modeling which captures the qualitative and, when possible, the quantitative behavior of systems with feedback relationships. Preferably, the symbolic model should be translatable into the differential equations which describe the behavior of the system being modeled.

(b) a reasoning method based on the symbolic modeling tool created in subgoal (a) which permits the inference of differential diagnoses (a set of hypothesized diagnoses) from incomplete data.

(c) a reasoning method based on subgoals (a) and (b) which permits inference of the state of the model for each hypothesized diagnosis. This subgoal would be satisfied by an algorithm which specifies a self-consistent set of values for all variables in the model, for a given hypothesis based on a given set of data. Such sets of data would constitute initial conditions for differential equations derived from the model.

(d) a simulation method, based on the model and its equivalent differential equations together with the initial conditions derived from the differential diagnosis (steps a-c above), for predicting the expected time course of the system being modeled for each hypothesized diagnosis. This method could also be used to predict the effects of treatments being considered for recommendation by the program.

(e) a reasoning method, based on domain-independent properties of the model, for shrinking and/or expanding the model automatically to use a minimal model configuration to account for normal and abnormal data.

(f) an explanation facility for examining the model, the given data, the inferred hypothesized diagnoses, predicted behaviors, and modifications of the model, to answer user queries and to teach fundamental concepts.

2. Cardiac Arrhythmia Recognition

The goals of this subproject are to develop:

(a) a symbolic model of the electrical system of the human heart, including pertinent anatomic and electrophysiologic features of the normal and diseased heart. The electrophysiologic features would include deterministic characteristics (e.g., conduction velocities, refractory periods), stochastic features (e.g., behavior of automatic foci), and temporal interactions (e.g., competing pacemakers).

(b) a symbolic/numeric representation of the observable features of the electrical activity of the heart, both surface ECG and intracardiac recordings, including noise. This representation would be intended to allow a feature extraction module working on actual patient data to communicate with a symbolic reasoning module, and would be translatable directly into waveform display format.

(c) a reasoning method for extracting features (identifying waveforms) from raw, digitized signal data. This method would augment established signal processing techniques by using knowledge-based algorithms to improve detection of P and T-U waves and to improve rejection of noise. It should be noted that this is itself a major research undertaking in the signal processing domain.

(d) a reasoning method for inferring the cardiac rhythms consistent with a given disease state in the model, similar to the prediction of consequences of the hemodynamic model in the first subproject. The output of this method would be in the symbolic/numerical representation of subgoal (b).

(e) a reasoning method for inferring possible disease states in the model from a given feature-extracted recording of the electrical activity of the heart. This subgoal constitutes cardiac arrhythmia interpretation, and is itself a major research project.

(f) a categorization method for inferring hierarchies of diagnoses from elementary abnormalities. For example, "periods of atrial fibrillation up to 30 minutes at up to 150 beats/min, supraventricular tachycardia of up to 10 beats length at a rate of 130 beats/min, and sinus bradycardia with a minimum rate of 45, all consistent with the sick sinus ("tachy-brady") syndrome" and "two QRS morphologies are present: they are narrow at rates less than 120 and are wide at rates above 120, consistent with a rate-dependent bundle branch block".

(g) an explanation facility for examining the model, the input data, and the interpretations to answer user queries and to teach fundamental concepts.

## *B. MEDICAL RELEVANCE AND COLLABORATIONS*

The two subprojects have related but separate medical goals:

1. Hemodynamic Modeling.

There are two subgoals in this subproject: model-based sensor integration and model-based care-giver assistance.

a. Model-based Sensor Integration

The long-range application of this subproject is the integration of patient-related data in the intensive care environment. Model- based real-time systems would allow the system to share a global understanding of the patient's condition with the human care-givers. Thus, it could interpret significant trends in key parameters and could draw attention to relationships which might otherwise escape attention in the constant flood of data common to these environments.

b. Model-based Care-giver Assistance

It could also serve as an assistant to the care-giver. In this mode, the human care-giver could evaluate the merits of proposed diagnostic and therapeutic measures in light of available data on the patient's condition.

Practical application of these concepts requires further development of the model and the reasoning algorithms, and extensive testing against real clinical scenarios. Refinement and quality control are presently the responsibility of the principal investigator, who is a board-certified cardiologist with subspecialty interest in cardiac electrophysiology.

Practical application also awaits general acceptance of standardized hospital data buses for automatic acquisition of important parameters now stored primarily on paper or on computers outside the intensive care setting, such as fluid inputs and outputs, medications, and results of invasive and non-invasive tests. Further, improved user interfaces will require better graphics and increased computer literacy on the part of care-givers.

2. Cardiac Arrhythmia Recognition.

The long-range application of this sub-project is in clinical devices such as intensive-care arrhythmia monitors, portable Holter monitors, and implantable cardioverter-defibrillators. There are two subgoals: recognition of surface electrocardiographic (ECG) recordings and recognition of intracardiac recordings.

a. Recognition of surface electrocardiographic recordings.

Substantial and well-recognized obstacles in signal processing will likely prevent non-AI algorithms from advancing beyond the current state of the art of interpretation of surface ECG recordings. These obstacles are primarily the problems of reliable detection of P and T-U waves, and rejection of noise.

We hope further that, by mimicking the behavior of expert human cardiologists, these obstacles can be bypassed if they cannot be overcome. We have enlisted as consultants Dr. William Long of M.I.T., who has long experience with modeling of the cardiovascular system and with symbolic approaches to ECG interpretation, and Dr. Charles Mead, a consultant formerly with the Division of Cardiology at Washington University - St. Louis who has 15 years experience with the development of algorithms for clinical arrhythmia recognition.

b. Recognition of intracardiac recordings.

Intracardiac recordings, which are taken from wires placed in the heart by percutaneous venous puncture or around the heart by surgery, are relatively free of P wave ambiguity and of noise. They are representative of the quality of signals available to implantable cardioverter-defibrillators.

Cardioverter-defibrillators are devices like pacemakers in that they monitor the heart rhythm in a patient to determine if an abnormality exists. They are capable of taking action (electrical countershock) if an appropriate abnormality is detected. Unlike ordinary pacemakers, these devices detect abnormalities characterized by rapid rates of heart activity, rather than excessively slow rates. They have been shown to reduce one-year mortality in high-risk patients from 30% to 2%, and they are expected to play an increasingly large role in treatment of such patients.

These relatively new devices currently use quite simple algorithms to detect abnormalities. The action they take consists of applying an electrical shock directly to the heart. This shock is frequently unpleasant to the patient. The problem is that the algorithms sometimes confuse innocent rapid heart rates, such as from exercise or atrial tachyarrhythmias, with lethal ventricular arrhythmias. This has proved troublesome enough to prompt repeated calls in the electrophysiology literature for improved algorithms for arrhythmia recognition in these devices.

The algorithms developed in this subproject would be suitable for this application when the computer power in the devices improves. Because these devices require powerful energy sources to perform repeated shocks over their lifetimes of 2-3 years, the power drain of more sophisticated computer chips is less important than it would be in ordinary pacemakers.

### *C. Highlights of Research Progress*

#### 1. Hemodynamic Modeling

Subgoals (a) through (d) have been accomplished in prototype form. The approach relies on a semi-quantitative representation (subgoal (a)) which assigns values by default if the user does not specify more detailed information. Constraint propagation using a dynamically generated semi-quantitative quantity space is performed by interpreting the model as a set of constraint equations. Domain- independent heuristics which recognize morphological features of the model are used to further constrain the propagation of constraints and to generate hypotheses when ambiguities arise. These heuristics generate a set of self-consistent hypotheses, each of which is a hypothesized diagnosis (subgoal b). Each hypothesized diagnosis is then refined by mathematical relaxation, in which the propagated values are treated as initial guesses, and the values are refined iteratively, again by interpreting the model as a set of constraint equations (subgoal c). In the several scenarios which have been examined, the value assignments achieved by hypothesis and iterative refinement have achieved correlation coefficients up to 0.90 with the values obtained by simulation of the same model. Phase (d) was accomplished by translating the model into a set of dynamical systems equations, which were then integrated in the standard manner.

We did not anticipate continuing this subproject after beginning the cardiac arrhythmia subproject. However, an opportunity has arisen to collaborate with Professor Benjamin Kuipers at the University of Texas at Austin on the development of algorithms which combine the advantages of classical qualitative simulation and of our semi- quantitative simulation approach. Professor Kuipers and a graduate student will spend the summer of 1988 in San Antonio for this purpose and to develop a case-based reasoning knowledge base of cardiovascular disorders.

2. Cardiac Arrhythmia Recognition

This subproject has been operational for almost one year. With support from the University of Texas Health Science Center at San Antonio and the American Heart Association, Texas Affiliate, we have acquired and integrated a Symbolics 3640 LISP workstation and an IBM- compatible microcomputer with attached optical scanner.

Progress has been made in two subprojects of this project:

a) Signal acquisition from paper records. Many standardized and interesting ECG recordings are available on paper but not in digitized form. To make these recordings machine-readable, we have developed software in TurboPascal (a popular microcomputer language) which digitizes the ECG signal on a paper ECG record by extracting the ECG line from the bit map of the image produced by the optical scanner. Algorithms have been developed with can extract the ECG line when grid lines are present and in the presence of varying intensity of the original image (as, for example, in xerox copies of paper records). The output of this software is a sequence of numbers representing the ECG voltage at sequential instants of time, equivalent to the output of an analog-to-digital converter. This set of numbers can then be used for development of ECG recognition software. In comparison to conventional analog-to-digital recordings, the optical scanner output has a root-mean-square error of 3.5-4.0% (Widman and Freeman), or about 1.5 times the thickness of a typical pen line on paper.

b) Identification of waveforms in raw signal data. Using digitized signals from electronic analog-to-digital conversion and the optical scanner conversion described above, we have implemented prototype software for waveform identification. It is organized by levels of abstraction. At the lowest level, conventional slope-analysis and filtering algorithms are used to identify the baseline and the QRS complexes. Knowledge-based slope-analysis algorithms then analyze the remaining signal to assign tentative labels of "P waves," "T waves," and "other" waves. Then, knowledge-based pattern recognition algorithms look for temporal and morphological relationships to confirm or correct the tentative labels. The prototype system is able to find P waves superimposed on T waves, a combination which cannot be detected by any commercial cardiac arrhythmia system.

### *D. List of Relevant Publications*

1. Widman, L.E. Knowledge-Based Fault Identification and "What If" Simulation in Symbolic Dynamic Systems Models. Proceedings of the 1988 Society for Computer Simulation Multiconference on AI and Simulation, pp. 89-94, 1988.

2. Widman, L.E., Lee, Y.-B., and Y.-H. Pao. Diagnosis of Causal Medical Models by Semi-Quantitative Reasoning. In: Miller, P.L. (ed.). Topics in Medical Artificial Intelligence, Springer-Verlag (in press)

3. Widman, L.E. Semi-Quantitative "Close Enough" Systems Dynamics Models: An Alternative to Qualitative Simulation. In: Widman, L.E., Loparo, K.A., and N.R. Nielsen (eds.). Artificial Intelligence, Simulation, and Modeling. John Wiley & Sons, New York (in press).

4. Widman, L.E., Loparo, K.A., and N.R. Nielsen (eds.). Artificial Intelligence, Simulation, and Modeling. John Wiley & Sons, New York (in press).

5. Widman, L.E. and G.L. Freeman. A-to-D Conversion from Paper Records with a Desktop Scanner and a Microcomputer. (submitted to the American Journal of Physiology).

*E. Funding Support*

1. American Heart Association, Texas Affiliate
Grant-in-Aid Award.
Knowledge-Based Computer Algorithms for Arrhythmia Analysis.
Principal Investigator: Lawrence E. Widman.
Award period: July, 1987 - June, 1988.
Level: $24,850 direct costs.

2. Medical Research Service, Department of Medicine and Surgery,
Veterans Administration.
Knowledge-Based Computer Algorithms for Cardiac Arrhythmia
Interpretation.
Principal Investigator: Lawrence E. Widman.
Award period: July, 1988 - June, 1990.
Level: $33,097, year 1; $26,647, year 2. (direct costs)

## II. INTERACTIONS WITH THE SUMEX-AIM RESOURCE

*A. Sharing and interactions with other SUMEX-AIM projects*

The major interaction with SUMEX-AIM this year has been communication with members of the AIM community, now that our work has been transferred to the Symbolics machine here.

SUMEX-AIM allows ready Email access to users of ARPANET, Bitnet and CSNet. This access has proved invaluable in communicating rapidly and effectively with co-workers at other institutions. The value of this access has been demonstrated particularly during the preparation of the book, Artificial Intelligence and Simulation, which will go into production in June, 1988. Email was used to communicate with contributors, to receive electronic copies of their manuscripts, and to issue unobtrusive, gentle but effective reminders when the manuscripts were late. Without Email access, the editorial process would have been much more difficult.

Review of the longer term history of this project shows that it would not exist had SUMEX-AIM not provided telecommunication support for the initial feasibility project in 1984-1985, which was carried out on the computers of the MIT Laboratory for Computer Science, Clinical Decision Making Group, and computational support during the development of the semi-quantitative approach in the years 1985-1987.

*C. Critique of Resource Management*

The service provided by SUMEX-AIM has been exemplary, largely because of prompt and effective response to difficulties as they arise. There has been a clear effort to assure that telecommunication access remained reliable during changes in commercial vendors, and the staff have responded to several technical questions promptly and accurately. Down-time has been minimal compared to that of other systems we have used, and is almost always scheduled several days in advance.

The reason we work within the AIM community is that it is the natural niche for our research interests. There is no short-term prospect that this project will reach commercial maturity or that it will lose sight of fundamental AI issues, and so we feel that it still belongs in the scientific AIM framework.

As noted in the previous section, the communication with other members of the AIM community has proved invaluable in the advancement of this project.

## III. RESEARCH PLANS

### *A. Project Goals and Plans*

The long range goals of this project are to develop intelligent comprehensive monitoring/alarm systems for intensive care unit settings; and intelligent arrhythmia recognition systems for monitors, Holter recorders, and implantable cardioverter/ defibrillators. The short term strategies for achieving these goals are discussed above.

The principle investigator is a staff cardiologist and Assistant Professor in the Division of Cardiology. He serves as an Attending Cardiologist during hemodynamic, angiographic and electrophysiological studies on selected patients. Substantial time is committed to research, and this project will constitute his major research emphasis.

### *B. Justification and requirements for continued SUMEX use*

The justification for this project is its potential for advancing the state of the art of expert system technology in the area of temporal reasoning and deep causal modeling, and for demonstrating practical use of expert symbolic computing in potentially life-saving, knowledge-intensive environments.

The requirements for continued SUMEX-AIM use should be the same as currently: telecommunications support, ARPANET access, about 3 megabytes of disc space, and a small amount of CPU time.

### *C. Needs and plans for Other Computing Resources Beyond SUMEX-AIM*

As predicted in last year's annual report, the main computational burden has been transferred to a local resource. SUMEX-AIM is needed primarily for communication and demonstration projects, as noted above.

### *D. Recommendations for Future Community and Resource Development*

Our strong recommendation is that SUMEX-AIM be maintained as a national AIM resource for communication, development of software useful to the AIM community, and sharing of demonstration projects. SUMEX-AIM could also serve as a central source of advice for new workstation users who may be geographically isolated from experienced workstation users.

Additionally, we would strongly support retention of the current telecommunication support and enough computing power to support promising young investigators who would otherwise not have access to symbolic computing power.

# IV.D.2. Pathfinder Project

**Pathfinder Project**

**Bharat Nathwani, M.D.**
**Department of Pathology**
**University of Southern California**

**Lawrence M. Fagan, M.D., Ph.D.**
**Department of Medicine**
**Stanford University**

## I. SUMMARY OF RESEARCH PROGRAM

### *A. Project Rationale*

Our project addresses difficulties in the diagnosis of lymph node pathology. Five studies from cooperative oncology groups have documented that, while experts show agreement with one another, the diagnosis made by practicing pathologists may have to be changed by expert hematopathologists in as many as 50% of the cases. Precise diagnoses are crucial for the determination of optimal treatment. To make the knowledge and diagnostic reasoning capabilities of experts available to the practicing pathologist, we have developed a pilot computer-based diagnostic program called Pathfinder. The project is a collaborative effort of the University of Southern California and the Stanford University Medical Computer Science Group. A pilot version of the program provides diagnostic advice on 72 common benign and malignant diseases of the lymph node based on 110 histologic features. Our research plans are to develop a full-scale version of the computer program by substantially increasing the quantity and quality of knowledge and to develop techniques for knowledge representation and manipulation appropriate to this application area. The design of the program has been strongly influenced by the INTERNIST/CADUCEUS program developed on the SUMEX resource.

Pathfinder computer science research is focused on the exploration and extension of formal techniques for decision making under uncertainty. Research foci include (1) the assessment and representation of important probabilistic dependencies among morphologic features and diseases, (2) the representation of knowledge about the progression of disease over time, (3) the acquisition and use of independent expert knowledge bases, (4) the customization of the system's reasoning and explanation behaviors to reflect the expertise of the user, and, (5) the explanation of complex formal reasoning techniques.

Toward the pragmatic goal of constructing a useful pathology teaching and decision support system, Pathfinder investigators are attempting to use intelligent computation to substantially increase the quantity and quality of pathology knowledge available to pathologists. Important areas of this knowledge integration task involve ongoing research on the crisp definition important morphologic features and feature severities, the synthesis of information from multiple experts, and the translation among multiple pathology classification schemes.

A group of expert pathologists from several centers in the U.S. have showed interest in the program and helped to provide the structure of the knowledge base for the Pathfinder system.

### *B. Medical Relevance and Collaboration*

One of the most difficult areas in surgical pathology is the microscopic interpretation of lymph node biopsies. Most pathologists have difficulty in accurately classifying lymphomas. Several cooperative oncology group studies have documented that while experts show agreement with one another, the diagnosis rendered by a "local" pathologist may have to be changed by expert lymph node pathologists (expert hematopathologists) in as many as 50% of the cases.

The National Cancer Institute recognized this problem in 1968 and created the Lymphoma Task Force which is now identified as the Repository Center and the Pathology Panel for Lymphoma Clinical Studies. The main function of this expert panel of pathologists is to confirm the diagnosis of the "local" pathologists and to ensure that the pathologic diagnosis is made uniform from one center to another so that the comparative results of clinical therapeutic trials on lymphoma patients are valid. An expert panel approach is only a partial answer to this problem. The panel is useful in only a small percentage (3%) of cases; the Pathology Panel annually reviews only 1,000 cases whereas more than 30,000 new cases of lymphomas are reported each year. A panel approach to diagnosis is not practical and lymph node pathology cannot be routinely practiced in this manner.

We believe that practicing pathologists do not see enough case material to maintain a high level of diagnostic accuracy. The disparity between the experience of expert hematopathology teams and those in community hospitals is striking. An experienced hematopathology team may review thousands of cases per year. In contrast, in a community hospital, an average of only ten new cases of malignant lymphomas are diagnosed each year. Even in a university hospital, only approximately 100 new patients are diagnosed every year.

Because of the limited numbers of cases seen, pathologists may not be conversant with the differential diagnoses consistent with each of the histologic features of the lymph node; they may lack familiarity with the complete spectrum of the histologic findings associated with a wide range of diseases. In addition, pathologists may be unable to fully comprehend the conflicting concepts and terminology of the different classifications of non-Hodgkin's lymphomas, and may not be cognizant of the significance of the immunologic, cell kinetic, cytogenetic, and immunogenetic data associated with each of the subtypes of the non-Hodgkin's lymphomas.

In order to promote the accuracy of the knowledge base development we will have participants for multiple institutions collaborating on the project. Dr. Nathwani will be joined by experts from Stanford (Dr. Dorfman), St. Jude's Children's Research Center -- Memphis (Dr. Berard) and City of Hope (Dr. Burke).

### *C. Highlights of Research Progress*

#### *C.1 Previous Accomplishments*

Since the project's inception in September, 1983, we have constructed several versions of Pathfinder. The first several versions of the program were *rule-based* systems like MYCIN and ONCOCIN which were developed earlier by the Stanford group. We soon discovered, however, that the large number of overlapping features in diseases of the lymph node would make a rule-based system cumbersome to implement. We next considered the construction of a *hybrid system*, consisting of a rule-based algorithm that would pass control to an INTERNIST-like scoring algorithm if it could not confirm the existence of classical sets of features. We finally decided that a modified form of the INTERNIST program would be most appropriate. The original version of Pathfinder is written in the computer language MacLisp and runs on the SUMEX DEC-20. This was transferred to Portable Standard Lisp (PSL) on the DEC-20, and later transferred to PSL on the HP 9836 workstations. Two M.D./Ph.D (Stanford Medical Information Science Program) students, David Heckerman and Eric Horvitz, designed and implemented the program and are continuing to lead research on the project.

The prototype knowledge base was constructed by Dr. Nathwani. During the early part of 1984, we organized two meetings of the entire team, including the pathology experts, to define the selection of diseases to be included in the system, and the choice of features to be used in the scoring process.

During the last three years, we have focused on methodologies for more accurately representing expert beliefs. In particular, we have used *influence diagrams* to represent dependencies among features in the Pathfinder knowledge base. A great deal of effort has been devoted to assessing and representing the intricate relationships among features that exist in the domain. We believe that this process will help to overcome some of the limitations of medical diagnostic systems.

We have also focused on the problem of complex information-theoretic inference. The explanation of a systems diagnostic behavior has been found to be of extreme importance to physicians. Unfortunately, it is often difficult to explain reasoning based on optimal models of inference. We have worked on the use of a set of alternative abstraction hierarchies to control inference. Our current techniques enable us to trade off optimality for the transparency of reasoning. We are now studying the control of this tradeoff to optimize inference.

We are also currently developing specialized probabilistic inference techniques that take advantage of particular patterns of independence that occur among features in lymph node pathology. In addition, we are developing richer mechanisms for controlling uncertain reasoning within the Pathfinder expert system.

### *C.2 The Pathfinder knowledge base*

The basic building block of the Pathfinder knowledge base is the disease profile or *frame.* Each disease frame consists of *features* useful for diagnosis of lymph node diseases. Currently these features include histopathologic findings seen in both low- and high-power magnifications. Each feature is associated with a list of exhaustive and mutually exclusive *values.* For example, the feature *pseudofollicularity* can take on any one of the values *absent*, *slight*, *moderate*, or *prominent*. These lists of values give the program access to *severity* information. In addition, these lists eliminate obvious interdependencies among the values for a given feature. For example, if pseudofollicularity is *moderate*, it cannot also be *absent*.

Qualitative dependencies among features for each disease are represented using the influence diagram methodology mentioned above. An influence diagram contains *nodes* and *arcs*. Nodes represent features and arcs represent dependencies among features. In particular, an arc is drawn from one feature to another when an expert believes that knowing one feature can change his beliefs that another feature will take on its possible values even when the diagnosis is known. Probabilities are used to quantitate the beliefs asserted by the expert.

### *C.3 Macintosh II Workstations*

We are currently moving the Pathfinder system into the Macintosh II environment. Thus the SUMEX-AIM resource will only be used for file archival and retrieval and for communication between the Stanford and USC Pathfinder research teams.

*D. Publications Since January 1984*

1. Horvitz, E.J., Heckerman, D.E., Nathwani, B.N. and Fagan, L.M.: *Diagnostic Strategies in the Hypothesis-directed Pathfinder System, Node Pathology.* HPP Memo 84-13. Proceedings of the First Conference on Artificial Intelligence Applications, Denver, Colorado, Dec., 1984.

2. Heckerman, D. E., and Horvitz, E. J., "The Myth of Modularity in Rule-based Systems," in Uncertainty in Artificial Intelligence, Vol. 2, J. Lemmer, L. Kanal, ed., North Holland, New York, 1987.

3. Horvitz, E.J., Heckerman, D.E., Nathwani, B.N. and Fagan, L.M.: *The Use of a Heuristic Problem-solving Hierarchy to Facilitate the Explanation of Hypothesis-directed Reasoning.* KSL Memo 86-2. Proceedings of MedInfo, Washington D.C., October, 1986.

4. Horvitz, E. J., "Toward a Science of Expert Systems," Invited Paper, Computer Science and Statistics: Proceedings of the 18th Symposium on the Interface, American Statistical Association, March, 1986, pgs. 45-52.

5. Heckerman, D.E., "An Axiomatic Framework for Belief Updates," in Uncertainty in Artificial Intelligence, Vol. 2, J. Lemmer, L. Kanal, ed., North Holland, New York, 1987.

6. * Heckerman, D. E., and Horvitz, E. J., "The Myth of Modularity in Rule-based Systems," in Uncertainty in Artificial Intelligence, Vol. 2, J. Lemmer, L. Kanal, ed., North Holland, New York, 1987.

7. * Heckerman, D.E., and Horvitz, E.J., "On the expressiveness of rule-based systems for reasoning under uncertainty," Proceedings of the National Conference on Artificial Intelligence, Seattle, Washington, July, 1987.

8. * Horvitz, E. J., Heckerman, D. E., Langlotz, C. P., "A framework for comparing alternative formalisms for plausible reasoning," Proceedings of the AAAI," August, 1986, Morgan Kaufman, Los Altos, CA, 1986,

9. * Horvitz, E.J., Breese, J.S., Henrion, M., Decision Theory in Expert Systems and Artificial Intelligence, International Journal of Approximate Reasoning, Elsevier, N.Y. July, 1988.

10. * Heckerman, D.E., An Evaluation of Three Scoring Schemes, Proceedings of the 4th AAAI Workshop on Uncertainty in Artificial Intelligence, Minneapolis, MN., (to appear August 1988).

11. * Horvitz, E.J., A Multiattribute Utility Approach to Inference Understandability and Explanation, Tech. Report, KSL-28-87, Knowledge Systems Laboratory, Stanford, California, March, 1987.

*E. Funding Support*

Research Grant submitted to National Institutes of Health
Grant Title: "Computer-aided Diagnosis of Malignant Lymph Node Diseases"
Principal Investigator: Bharat Nathwani
Funding for three years from the National Library of Medicine
1 RO1 LM 04529

$766,053 (direct and indirect)

Professional Staff Association, Los Angeles County Hospital, $10,000.

University of Southern California, Comprehensive Cancer Center, $30,000.

Project Socrates, Univ. of Southern Calif., Gift from IBM of IBM PC/XT.

## II. INTERACTIONS WITH THE SUMEX-AIM RESOURCE

### *A. Medical Collaborations and Program Dissemination via SUMEX*

Because our team of experts are in different parts of the country and the computer scientists are not located at the USC, we envision a tremendous use of SUMEX for communication, demonstration of programs, and remote modification of the knowledge base. The proposal mentioned above was developed using the communication facilities of SUMEX.

### *B. Sharing and Interaction with Other SUMEX-AIM Projects*

Our project depends heavily on the techniques developed by the INTERNIST/CADUCEUS project. We have been in electronic contact and have met with members of the INTERNIST/CADUCEUS project, as well as been able to utilize information and experience with the INTERNIST program gathered over the years through the AIM conferences and on-line interaction. Our experience with the extensive development of the pathology knowledge base utilizing multiple experts should provide for intense and helpful discussions between our two projects.

The SUMEX pilot project, RXDX, designed to assist in the diagnosis of psychiatric disorders, is currently using a version of the Pathfinder program on the DEC-20 for the development of early prototypes of future systems.

### *C. Critique of Resource Management*

The SUMEX resource has provided an excellent basis for the development of a pilot project. The availability of a pre-existing facility with appropriate computer languages, communication facilities (especially the TELENET network), and document preparation facilities allowed us to make good progress in a short period of time. The management has been very useful in assisting with our needs during the start of this project.

## III. RESEARCH PLANS

### *A. Project Goals and Plans*

*Collection and refinement of knowledge about lymph node pathology*

The knowledge base of the program is about to undergo revision by the experts, and then will be extensively tested. We will be extending the knowledge base to capture important probabilistic dependencies among features. We will also begin to evaluate the knowledge base and inference techniques.

Other possible extensions include: developing techniques for simplifying the acquisition and verification of knowledge from experts, and creating mapping schemes that will facilitate the understanding of the many classifications of non-Hodgkin's lymphomas. We will also attempt to represent knowledge about special diagnostic entities, such as multiple discordant histologies and atypical proliferations, which do not fit into the classification methods we have utilized.

*Representation Research*

We have been working to enhance the knowledge base by structuring features so that overlapping features are not incorrectly weighted in the decision making process, implementing new methods for scoring hypotheses, and creating appropriate explanation capabilities.

### *B. Requirements for Continued SUMEX Use*

We are currently dependent on the SUMEX computer for file storage and archival, and for communication. We have transferred the program over to Portable Standard Lisp which is used by several users on the SUMEX system. While the switch to workstations has lessened our requirements for computer time for the development of the algorithms, we will continue to need the SUMEX facility for the interaction with each of the research locations specified in our NIH proposal. The SUMEX community version of the program is stored on the SUMEX mainframe for use by non-Stanford users.

### *C. Requirements for Additional Computing Resources*

Most of our computing resources will be met by the 2060 plus the use of the Macintosh II workstations. We will need additional file space on the 2060 as we quadruple the size of our knowledge base through the construction of multiple knowledge bases. We will continue to require access to the 2060 for communication purposes, access to other programs, and for file storage and archiving.

### *D. Recommendations for Future Community and Resource Development*

We encourage the continued exploration by SUMEX of the interconnection of workstations within the mainframe computer setting. We will need to be able to quickly move a program from workstation to workstation, or from workstation back and forth to the mainframe. Software tools that would help the transfer of programs from one type of workstation to another would also be quite useful. Until the type of workstations that we are using in this research becomes inexpensive, we will continue to need a machine like SUMEX to provide others with a chance to experiment with our software.

# IV.D.3. Knowledge Engineering for Radiation Therapy

## KNOWLEDGE ENGINEERING FOR RADIATION THERAPY

**Ira J. Kalet, Ph.D.**
**Witold Paluszinski**
**University of Washington**
**Seattle, Washington**

### I. Summary of Research Program

*A. Project Rationale*

We are developing an expert system for planning of radiation therapy for head and neck cancers. The project will ultimately combine knowledge-based planning with numerical simulation of the radiation treatments. The numerical simulation is needed in order to determine if the proposed treatment will conform to the goals of the plan (required tumor dose, limiting dose to critical organs). The space of possible radiation treatments is numerically very large, making traditional search techniques impractical. Yet, with modern radiation therapy equipment, the design of treatment plans might be significantly aided by automatically generating plans that meet the treatment constraints. The project will result in systematization of knowledge about radiation treatment design, and will also provide an example of how to represent and solve design problems with a knowledge based system.

*B. Medical Relevance and Collaborations*

Radiation therapy has shown dramatic improvement in the cure rate for many tumor sites in the last two decades. Much of this can be attributed to the improved penetration capability of modern megavoltage X-ray machines. These high energy beams can deliver high tumor doses without overdosing surrounding tissue in many cases. However, they are typically used in very limited ways, because of the lack of suitable simulation systems to compute the dose distribution for any but a few narrow choices of treatment geometry. In the last few years these simulation systems have been extended to the full range of geometric treatment arrangement that any therapy machine is capable of. Thus it would be valuable to be able to generalize our knowledge of treatment technique by exploring these expanded possibilities. In addition, even treatments with standard geometries can be very complex, and it is tedious to explore all of them individually. A knowledge-based system can generate a few "best" plans which satisfy the constraints and allow more time for the physician to evaluate the options, or make minor adjustments for optimization.

Since cancer treatment is a multi-disciplinary approach involving surgery and chemotherapy as well as radiation, it is important to coordinate this work with knowledge-based program projects in those areas. Most significant is the ONCOCIN project, which addresses management of patients on chemotherapy protocols.

This project has some relevance to computer science as well, in that our approach, if successful, may contribute to a better understanding of design problem solving with knowledge-based systems.

## C. Highlights of Research Progress

In the past year, we have made further additions to the rule database for details of head and neck cancer treatment. In addition we have examined the issues of control strategy associated with using prototypes in planning. The major effort has been to rewrite the entire program in COMMON LISP on a VAXstation II/GPX, and to use the Portable Common LOOPS (PCL) package, which appears to be on the way to becoming a standard (Common Lisp Object System - CLOS). Following the successful rewrite, we have added code to allow the lisp program to run the dose computation program as a subprocess, and exchange messages with that subprocess. The expert system can now create a plan, run the dose computation and receive results, analyzed into a form that will facilitate plan refinement.

Our expert system now has about two hundred rules, a two-level (agenda-based) control strategy, and about ten prototypes for plan construction. It is now implemented in COMMON LISP, on a VAX running the VMS operating system. This environment was chosen because it is also the environment used for a graphic simulation system that does radiation dose calculations for arbitrary treatment plans. The dose calculation is needed to determine whether a plan meets the treatment goals set by the system in its early phases of planning.

## D. List of Relevant Publications

1. I. Kalet and W. Paluszynski: A Production Expert System for Radiation Therapy Planning. Proceedings of the AAMSI Congress 1985, May 20-22, 1985, San Francisco, California. Edited by Allan H. Levy and Ben T. Williams. American Association for Medical Systems and Informatics, Washington, D.C., 1985.

2. W. Paluszynski and I. Kalet: Radiation Therapy Planning: A Design Oriented Expert System. WESTEX-87 (Western Conference on Expert Systems), Anaheim, California, June 2-4, 1987.

3. I. Kalet and J. Jacky: Knowledge-based Computer Simulation for Radiation Therapy Planning. Proceedings of the Ninth International Conference on the use of Computers in Radiotherapy, Scheveningen, the Netherlands, June 1987. North Holland, 1987.

4. I. Kalet and W. Paluszynski: Knowledge-based Computer Systems for Radiotherapy Planning. American Journal of Clinical Oncology. In press.

## II. Interactions with the SUMEX-AIM Resource

Our main use of the SUMEX-AIM resource has been as a means to be in contact with other researchers working on AIM projects. The existence of a mailbox at SUMEX-AIM has made it much easier for colleagues at other institutions to communicate with us, and has been valuable in assisting us with organizing the AIM Workshop for 1987.

We have had a great deal of contact with members of the ONCOCIN project and other groups. This has been valuable to us in stimulating creative approaches to our project.

## III. Research Plan

### *A. Project Goals and Plans*

We plan to continue to acquire rules and develop our current expert system. This includes solving problems of use of prototypes, satisfaction of constraints by some kind of backtracking search, and incorporating evaluation of plans by using the results of the dose computation. This last idea involves coupling the expert system with the dose computation system (written in PASCAL) in suitably efficient ways. We are implementing rules for plan improvement by modification of parameters, and by adding or deleting treatment plan components (treatment fields). Our long-term goal is to shape the user interface and improve the system performance to where it can provide assistance to clinicians in treatment design for patients in the normal course of treatment.

### *B. Justification and Requirements for Continued SUMEX use*

We foresee continued need to be in touch with other members of the AIM community, particularly projects centered at SUMEX. While we do not expect to use the computing resources of SUMEX directly, some more extensive communication and involvement is likely to be useful.

### *C. Plans For Other Computing Resources*

The main computing resources for our project will continue to be local. We continue to develop the expert system code in VAX Lisp, an implementation of Common Lisp on the DEC VAXstation. We are currently using a VAXstation II/GPX. This appears to be a good choice to satisfy our need for high performance graphic simulation and a reasonable Lisp system. However, the resources for the dose computation may not be adequate as we incorporate more sophisticated computation models. As this develops, we hope to experiment with distributed systems, in which the dose computation may run on a remote resource, which may or may not be at SUMEX.

### *D. Recommendations for Future Community and Resource Development*

Two areas will be of increasing importance to us in the future: communication capabilities (electronic mail and file transfer) and centralized databases. By centralized databases, we refer to the need for better maintenance of mailing lists, information about projects, and possibly on-line reports. Dr. Kalet's experience in organizing the AIM Workshop for 1987 demonstrated that electronic communication is invaluable, even in its present state, but in order to create a list to send announcements to, we expended many hours of manually cutting and pasting messages containing past lists and searching for up-to-date electronic mail addresses.

If fees for use of SUMEX resources were imposed, the main impact on our project would be one of increased isolation, unless we could find grant support for the fees.

## IV.E. AIM Communications Users

One of the key functions of the SUMEX-AIM resource has been inter-community communications. As the CPU-cycle-provider part of SUMEX's role phases out according to our plan for the distributed AIM community, this communications role will become increasingly important. In a recent action, the AIM Executive Committee approved broad guidelines for admitting communications users to SUMEX-AIM and we are implementing that policy.

Over the past year, several of the projects that had been using SUMEX-AIM as their primary source of computing resources have obtained local resources and now use SUMEX only for communications contact with others in the AIM community. Those projects include:

- *Hierarchical Models of Human Cognition (CLIPR)*, under Professors Walter Kintsch and Peter Polson at the University of Colorado.
- *Problem Solving Expertise (SOLVER)*, under Professors Paul Johnson and William Thompson at the University of Minnesota.
- *RXDX*, under Professor Robert Lindsay at the University of Michigan.
- *Computer-Based Exercises in Pathophysiologic Diagnosis*, under Professor J. Robert Beck at Dartmouth College.
- *SOAR*, under Dr. Paul Rosenbloom at USC/ISI in Los Angeles (formerly Stanford University).
- *Logic Group projects (DART, Intelligent Agents, and MRS)*, under Professor Michael Genesereth at Stanford University

Other projects will be undergoing similar transitions this coming year and we will report these in the next annual report.

The following is a list of users and groups from the National AIM community who use the SUMEX-AIM resource primarily for communication. All of the people listed have logged on to SUMEX over the past report year or have their mail forwarded to other sites. Many have been investigators on past projects that made use of SUMEX as a CPU provider and are now doing their research computing on machines at their home institutions. No detailed reports on their research are included.

| | |
|---|---|
| <AAAI-OFFICE> | American Association for Artificial Intelligence<br>(AAAI Hdqtrs.) |
| <ALTENBERG> | ALTENBERG, Lee, Ph.D.<br>Department of Biochemistry<br>Stanford University Medical Center |
| <AMAREL> | AMAREL, Saul, Ph.D.<br>Department of Computer Science<br>Rutgers University |
| <ANZALDI> | ANZALDI, Emmanuele<br>Instituto di Informatica e Sistemistica<br>Universita' di Pavia, Italy |

| | |
|---|---|
| <BAKER> | BAKER, William R., Jr., Ph.D.<br>Knowledge Research Associates<br>Bradenton, Fl. 34203-0035 |
| <BECK> | BECK, J. Robert, M.D.<br>Dartmouth College, School of Medicine |
| <BELLAIRS> | BELLAIRS, Keith<br>University of Minnesota |
| <BRAND> | BRAND, Tony<br>Chemistry Department<br>Trenton State College, NJ |
| <CHANDRASEKARAN> | CHANDRASEKARAN, B., Ph.D.<br>Department of Computer and Information Science<br>Ohio State University |
| <DARDEN> | DARDEN, Lindley, Ph.D.<br>Committee on the History and<br>Philosophy of Science<br>University of Maryland |
| <DAVIS> | DAVIS, Randall, Ph.D.<br>Massachusetts Institute of Technology |
| <DAVISON> | DAVISON, Daniel, B., Ph.D.<br>Department of Biochemical and Biophysical Sciences<br>University of Houston |
| <DIETTERICH> | DIETTERICH, Tom, Ph.D.<br>Department of Computer Science<br>Oregon State University |
| <DROGERS> | ROGERS, David<br>Research Institute for Advanced Computer Science<br>NASA Ames Research Center |
| <DUMAS> | DUMAS, Jean-Pierre, Ph.D.<br>France |
| <FELTOVICH> | FELTOVICH, Paul J.<br>Southern Illinois University School Of Medicine |
| <FREKSA> | FREKSA, Christian<br>Institut fuer Informatik<br>Technische Universitaet Muenchen, West Germany |
| <FRISSE> | FRISSE, Mark, M.D.<br>Department of Internal Medicine<br>Washington University School of Medicine |
| <GOERZ> | GOERZ, Gunther<br>Friedrich-Alexander-Universitat<br>Erlangen, West Germany |

| | |
|---|---|
| <HAMM> | HAMM, Greg H.<br>European Molecular Biology Laboratory<br>Heidelberg, W. Germany |
| <HEDRICK> | HEDRICK, Charles L., Ph.D.<br>Center for Computer and Information Services<br>Rutgers University |
| <HENRION> | HENRION, Max<br>Carnegie-Mellon University |
| <HORSWILL> | HORSWILL, Ian D.<br>University of Minnesota |
| <HUMPHREY> | HUMPHREY, Susanne M.<br>National Library of Medicine |
| <ICRF> | EXOMOLGEN Project Directory<br>c/o Walter BODMER, Ph.D.<br>Imperial Cancer Research Fund Laboratories<br>London, England |
| <JARONSON> | ARONSON, Jules<br>National Library of Medicine |
| <KAIHARA> | KAIHARA, Shigekoto, M.D.<br>University of Tokyo Hospital<br>Tokyo, JAPAN |
| <KDUNCAN> | DUNCAN, Karen<br>Health Information Systems<br>Los Altos, California |
| <KERN> | KERN, Kevin B.<br>Lab for Computer Science Research<br>Rutgers University |
| <KINGSLAND> | KINGSLAND, Lawrence C., III, Ph.D.<br>National Library of Medicine |
| <KULIKOWSKI> | KULIKOWSKI, Casimir, Ph.D.<br>Department of Computer Science<br>Rutgers University |
| <LAGADIC> | LAGADIC Pascal<br>Clinique Kerfriden<br>Chateaulin, France |
| <LANGRIDGE> | LANGRIDGE, Robert, Ph.D.<br>Computer Graphics Laboratory<br>University of California at San Francisco |
| <LEDERBERG> | LEDERBERG, Joshua, Ph.D.<br>The Rockefeller University |

| | |
|---|---|
| <LESGOLD> | LESGOLD, Alan M., Ph.D.<br>Learning Research & Development Center<br>University of Pittsburgh |
| <LINDBERG> | LINDBERG, Donald, M.D.<br>National Library of Medicine |
| <MASINTER> | MASINTER, Larry<br>Xerox Corporation, Palo Alto Research Center |
| <MAXAM> | Allan Maxam, Ph.D.<br>Harvard Medical School |
| <MAZZETTI> | MAZZETTI, Claudia<br>Executive Director, American Association<br>for Artificial Intelligence |
| <MCDONALD> | MCDONALD, Clement J.<br>University of Indiana Medical Center |
| <MKENT> | KENT, Marty<br>Learning Research & Development Center<br>University of Pittsburgh |
| <MOEN> | MOEN, James<br>Center for Research in Human Learning<br>University of Minnesota |
| <NOVAK> | NOVAK, Gordon, Ph.D.<br>Computer Science Department<br>University of Texas at Austin |
| <PEARSON> | PEARSON, William R., Ph.D.<br>Department of Biochemistry<br>University of Virginia |
| <PJOHNSON> | JOHNSON, Paul E., Ph.D.<br>Center for Research in Human Learning<br>University of Minnesota |
| <POLSON> | POLSON, Peter G., Ph.D.<br>Department of Psychology<br>University of Colorado |
| <RGREENES> | GREENES, Robert, M.D.<br>Department of Radiology<br>Harvard Brigham and Women's Hospital |
| <SIEBER> | SIEBER, Willi, Ph.D.<br>Sandoz Ltd., Switzerland |
| <SLAGLE> | SLAGLE, James<br>University of Minnesota |

| | |
|---|---|
| <SOTOS> | SOTOS, John<br>Division of Cardiology<br>Johns Hopkins Hospital |
| <STEFIK> | STEFIK, Mark, Ph.D.<br>Xerox Palo Alto Research Center |
| <SUWA> | SUWA, Motoi<br>Tsukuba, JAPAN |
| <SZOLOVITS> | SZOLOVITS, Peter<br>Massachusetts Institute of Technology |
| <THOMPSON> | THOMPSON, William B., Ph.D.<br>Computer Science Department<br>University of Minnesota |
| <VANBEMMEL> | VAN BEMMEL, Jan<br>Vrije Universiteit<br>Amsterdam, Netherlands |
| <WEBBER> | WEBBER, Bonnie<br>Dept. of Computer and Information Sciences<br>University of Pennsylvania |
| <WEISS> | WEISS, Sholom, Ph.D.<br>Department of Computer Science<br>Rutgers University |
| <WIPKE> | WIPKE, W. Todd, Ph.D.<br>Department of Chemistry<br>University of California at Santa Cruz |
| <ZINK> | ZINK, Sandra, Ph.D.<br>Lawrence Berkeley Labs |
| <ZUKER> | ZUKER, Michael, Ph.D.<br>National Research Council<br>Ottawa, Canada |

# Appendix A

## Knowledge Systems Laboratory Brochure

# ARTIFICIAL INTELLIGENCE RESEARCH IN THE KNOWLEDGE SYSTEMS LABORATORY

**Stanford University**
**Department of Computer Science**
**Department of Medicine**

**January 1988**

## Introduction

The Knowledge Systems Laboratory (KSL) is an artificial intelligence (AI) research laboratory of over 100 people—faculty, staff, and students—within the Departments of Computer Science and Medicine at Stanford University. KSL is the new name for the interdisciplinary AI research community that has evolved over the past two decades. Begun as the DENDRAL Project in 1965 and known as the Heuristic Programming Project from 1972 to 1984, the new organization reflects the diversity of the research now under way. The KSL is a modular laboratory, consisting of four collaborating yet distinct groups with different research themes:

- **The Heuristic Programming Project (HPP)**, Professor Edward A. Feigenbaum, scientific director—large, multi-use knowledge bases, blackboard systems, concurrent system architectures for AI, automated software design, expert systems for science and engineering. Executive director: Robert Engelmore. Research scientists: Harold Brown, Scott Clearwater, Bruce Delagi, Barbara Hayes-Roth, Hirotoshi Maegawa, H. Penny Nii, and Hiroshi Okuno.

- **The HELIX Group**, Professor Bruce G. Buchanan, scientific director—machine learning, transfer of expertise, and problem solving. Research scientists: James Brinkley, William J. Clancey, Craig Cornelius, Diana Forsythe, Barbara Hayes-Roth, Rich Keller, Catherine Manago.

- **The Medical Computer Science (MCS) Group**, Associate Professor Edward H. Shortliffe, scientific director (Department of Medicine with courtesy appointment in Computer Science)—fundamental research and advanced biomedical applications in the area of AI and decision sciences; includes the Medical Information Sciences (MIS) program. Assistant Professor: Mark A. Musen; Research scientists: Gregory F. Cooper, Lawrence M. Fagan (Associate Director).

- **The Symbolic Systems Resources Group (SSRG)**, Thomas C. Rindfleisch, scientific director (joint appointment Departments of Computer Science and Medicine)—research on and operation of distributed computing resources for AI research, including the SUMEX-AIM facility. Assistant director: William J. Yeager.

The KSL is guided by an Executive Committee consisting of the four sublaboratory directors. Tom Rindfleisch serves as overall KSL director.

This brochure summarizes the goals and methodology of the KSL, its research and academic programs, its achievements, and the research environment of the laboratory.

## Basic Research Goals and Methodology

Throughout a 20-year history, the KSL and its predecessors, DENDRAL and HPP, have concentrated on research in expert systems—that is, systems using symbolic reasoning and problem-solving processes that are based on extensive domain-specific knowledge. The KSL's approach has been to focus on applications that are themselves significant real-world problems, in domains such as science, medicine, engineering, and education, and that also expose key, underlying AI research issues. For the KSL, AI is largely an empirical science. Research problems are explored, not by examining strictly theoretical questions, but by designing, building, and experimenting with programs that serve to test underlying theories.

The basic research issues at the core of the KSL's interdisciplinary approach center on the computer representation and use of large amounts of domain-specific knowledge, both factual and heuristic (or judgmental). These questions have guided our work since the 1960s and are now of central importance in all of AI research:

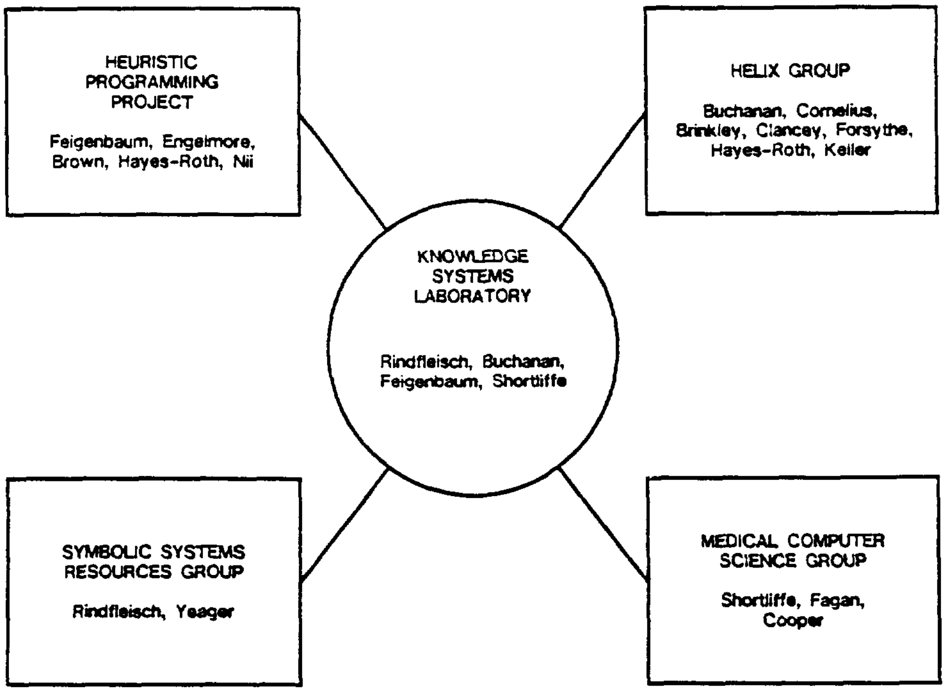
1. **Knowledge representation.** How can the knowledge necessary for complex problem solving be represented for its most effective use in automatic inference processes? Often, the knowledge obtained from experts is heuristic knowledge, gained from many years of experience. How can this knowledge, with its inherent vagueness and uncertainty, be represented and applied? How can knowledge be represented so that it can be used for many problem solving purposes?

2. **Knowledge acquisition.** How is knowledge acquired most efficiently—whether from human experts, from observed data, from experience, or by discovery? How can a program discover inconsistency and incompleteness in its knowledge base? How can knowledge be added without perturbing the established knowledge base unnecessarily?

3. **Use of knowledge.** By what inference methods can many sources of knowledge of diverse types be made to contribute jointly and efficiently toward solutions? How can knowledge be used intelligently, especially in systems with large knowledge bases, so that it is applied in an appropriate manner at the appropriate time?

4. **Explanation and tutoring.** How can the knowledge base and the line of reasoning used in solving a particular problem be explained to users? What constitutes a sufficient or an acceptable explanation for different classes of users?

5. **System tools and architectures.** What kinds of software tools and system architectures can be constructed to make it easier to implement expert programs with greater complexity and higher performance? What kinds of systems can serve as vehicles for the cumulation of knowledge of the field for the researchers?


HEURISTIC PROGRAMMING PROJECT
Feigenbaum, Engelmore, Brown, Hayes-Roth, Nii
HELIX GROUP
Buchanan, Cornelius, Brinkley, Clancey, Forsythe, Hayes-Roth, Keller
KNOWLEDGE SYSTEMS LABORATORY
Rindfleisch, Buchanan, Feigenbaum, Shortliffe
SYMBOLIC SYSTEMS RESOURCES GROUP
Rindfleisch, Yeager
MEDICAL COMPUTER SCIENCE GROUP
Shortliffe, Fagan, Cooper

**Knowledge Systems Laboratory Organization**

## Current Research Projects

The following list of projects now under way within the four KSL research groups gives a brief summary of the major goals of each project and lists the personnel (staff and Ph.D. candidates) directly involved. More complete information on individual projects can be obtained from the person indicated as the project contact. Inquiries should be addressed in care of:

Knowledge Systems Laboratory
Department of Computer Science
Stanford University
701 Welch Road, Building C
Palo Alto, CA 94304
415-723-3444

### The Heuristic Programming Project

- **Advanced Architectures Project**—Design a new generation of computer architectures to exploit concurrency in blackboard-based signal understanding systems.
  *Personnel:* Edward A. Feigenbaum (contact), Nelleke Aiello, Harold Brown, Bruce Delagi (DEC), Robert Engelmore, Hirotoshi Maegawa (Sony), Penny Nii, Sayuri Nishimura, Hiroshi Okuno (NTT), James Rice, Nakul Saraiya.

- **Blackboard Architecture Project**—Integrate current knowledge about blackboard framework problem-solving systems and develop a domain-independent model that includes knowledge-based control processes.
  *Personnel:* Barbara Hayes-Roth (contact), Micheal Hewett, Penny Nii.

- **Large Multi-use Knowledge Bases (LMKB)**—Develop a knowledge base of scientific and engineering facts, principles and methods, along with appropriate representations of the knowledge, for multiple uses, including diagnosis and monitoring, planning, configuration, and tutoring.
  *Personnel:* Edward Feigenbaum (contact), Richard Keller, Scott Clearwater (LANL), Robert Engelmore.

- **Automated Software Design**—Assist software designers in designing new program modules via intelligent selection and modification from a library of existing software modules.
  *Personnel:* Penny Nii(contact), Cordell Green (Kestrel Institute).

### The HELIX Group

- **PROTEAN**—Study complex symbolic constraint-satisfaction problems in the blackboard framework with application to protein structure determination from nuclear magnetic resonance data.
  *Personnel:* Bruce Buchanan (contact), Oleg Jardetzky (Stanford Magnetic Resonance Laboratory), Russ Altman, Jim Brinkley, Enrico Carrara, Craig Cornelius, Bruce Duncan, Guido Haymann-Haber, Olivier Lichtarge.

- **NEOMYCIN/GUIDON2**—Develop knowledge representation and explanation capabilities for computer-aided teaching of diagnostic reasoning. This work is moving to the Xerox Institute for Research on Learning in Spring 1988.
  *Personnel:* Bill Clancey (contact), Stephen Barnhouse, Bob London, Steve Oliphant.

- **Knowledge Acquisition Studies**—Study the processes for transferring knowledge into a computer program, including learning by induction, analogy, watching, chunking, reading, and discovery.
  *Personnel:* Bruce Buchanan (contact), Martin Chavez, Tze-Pin Cheng, Diana Forsythe, Haym Hirsh, Richard Keller, Harold Lehmann, Eric Schoen, John Sullivan.

- **Financial Resources Management**—Develop a constraint-based expert system for financial resource planning.
  *Personnel:* Bruce Buchanan and Tom Rindfleisch (contacts), Craig Cornelius, Andy Gelman, Catherine Manago.

- **Large Multi-use Knowledge Bases (LMKB)**—See description under HPP.

**The Medical Computer Science Group**

- **ONCOCIN**—Develop knowledge-based systems for the administration of complex medical treatment protocols such as those encountered in cancer chemotherapy.
  *Personnel:* Ted Shortliffe (contact), Charlotte Jacobs (Oncology), Larry Fagan, David Combs, Robert Carlson, Christopher Lane, Curt Langlotz, Rick Lenon, Mark Musen, Janice Rohn, Samson Tu, Cliff Wulfman, Andrew Zelenetz.

- **OPAL/PROTEGE**—Develop graphics-based knowledge acquisition tools for clinical trials. OPAL developed out of the ONCOCIN project to provide a method for specifying cancer treatment experiments. The PROTEGE program is capable of creating OPAL-like knowledge acquisition tools for various areas of medicine.
  *Personnel:* Mark Musen (contact), Larry Fagan, Ted Shortliffe, David Combs, Eric Sherman.

- **Speech Input to Expert Systems**—Develop multi-modal interface to expert systems, concentrating on a connected speech input device. Primary application will be extension to the ONCOCIN graphical interface.
  *Personnel:* Larry Fagan (contact), Bonnie Webber (University of Pennsylvania), Ted Shortliffe, Ed Feigenbaum (HPP), Ellen Isaacs (Psycholinguistics), Clifford Wulfman.

- **Physician's Workstation**—Develop advanced integrated workstation suitable for providing decision support functions to clinicians in both inpatient and outpatient settings; initial work in the area of cardiovascular disease prevention, with an emphasis on the management of lipid disorders.
  *Personnel:* Ted Shortliffe (contact), John Schroeder (Cardiology), David Maron (Heart Disease Prevention Center), Jonathan King, Tom Rindfleisch, Don Rucker, Joan Walton.

- **Blackboard/Intensive Care Unit (BBICU)**—Interpret data from the intensive care unit and suggest therapy plans for patients with mechanical breathing support. Two aspects of the project are: (1) representing the structure and function of the body and (2) combining qualitative and quantitative reasoning techniques.
  *Personnel:* Larry Fagan (contact for qualitative/quantitative), Barbara Hayes-Roth (HPP - contact for structure/function), Adam Seiver (Palo Alto Veterans Hospital), Lewis Sheiner (University of California, San Francisco), Ingo Beinlich, Reed Hastings, Micheal Hewett, Noi Hewett, Michael Kahn (UCSF), Nick Parlante (Palo Alto VA Hospital), John Reed, George Thomsen, Rich Washington.

- **Probabilistic Expert Systems**—Develop pragmatic and theoretically sound methods for the acquisition and computation of probabilistic information within medical expert systems.
  *Personnel:* Greg Cooper (contact), Ted Shortliffe, David Heckerman, Eddie Herskovits, Eric Horvitz, Jaap Suermondt.

**The Symbolic Systems Resources Group (SSRG)**

- **SUMEX-AIM Resource**—Develop and operate a national computing resource for biomedical applications of artificial intelligence in medicine and for basic research in AI at KSL. *Personnel*: Tom Rindfleisch (contact), Rich Acuff, Mark Crispin, Frank Gilmurray, Michael Marria, Christopher Schmidt, Andrew Sweer, Bob Tucker, Nicholas Veizades, Bill Yeager.
- **AI Workstation and Network Systems**—Develop network-based computing environments for Lisp workstations including remote graphics and distributed computing. *Personnel*: SSRG staff
- **Financial Resources Management**—See description under HELIX.

## Students and Special Degree Programs

Graduate students are an essential part of the research productivity of the KSL. Currently 36 students are working with our projects centered in Computer Science and another 21 students are working with the MCS/MIS programs in Medicine. Of the 36 working in Computer Science, 16 are working toward Ph.D. degrees, and 20 are working toward M.S. degrees. A number of these students are pursuing interdisciplinary programs and come from the Departments of Engineering, Mathematics, Education, and Medicine. Of the 21 working in Medicine, 15 are working toward Ph.D. degrees, and 6 are working toward M.S. degrees.

Because of the highly interdisciplinary and experimental nature of KSL research, two special degree programs have been established:

**Medical Information Sciences (MIS)**— an interdepartmental program approved by Stanford University in 1982. It offers instruction and research opportunities leading to the M.S. or Ph.D. degree in medical information sciences, with an emphasis on either medical computer science or medical decision science. The program, directed by Ted Shortliffe and co-directed by Larry Fagan, is formally administered by the School of Medicine, but the curriculum and degree requirements are coordinated with the Dean of Graduate Studies and the Graduate Studies Committee of the University. The program reflects our local interest in the interconnections between computer science, artificial intelligence, and medical problems. Emphasis is placed on providing trainees with a broad conceptual overview of the field and with an ability to create new theoretical and practical innovations of clinical relevance.

**Master of Science in Computer Science: Artificial Intelligence (MS:AI)**— a terminal professional degree offered for students who wish to develop a competence in the design of substantial knowledge-based AI applications but who do not intend to obtain a Ph.D. degree. The MS:AI program is administered by the Committee for Applied Artificial Intelligence, composed of faculty and research staff of the Computer Science Department. Normally, students spend two years in the program with their time divided equally between course work and research. In the first year, the emphasis is on acquiring fundamental concepts and tools through course work and project involvement. During the second year, students implement and document a substantial AI application project.

## Academic and Research Achievements

The primary products of our research are scientific publications on the basic research issues that motivate our work, computer software in the form of the expert systems and AI architectures we develop, and the students we graduate who continue AI research in other academic and industrial laboratories.

The KSL has averaged publishing more than 45 research papers per year in the AI literature, including journal articles, theses, proceedings articles, and working papers.* In addition, many talks and invited lectures are given annually. In the past few years, 11 major books have been published by KSL faculty, staff, and former stndents, and several more are in progress. Those recently published include:

- *Heuristic Reasoning about Uncertainty: An AI Approach,* Cohen, Pitman, 1985.
- *Readings in Medical Artificial Intelligence: The First Decade,* Clancey and Shortliffe, Addison-Wesley, 1984.
- *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Buchanan and Shortliffe, Addison-Wesley, 1984.
- *The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World*, Feigenbaum and McCorduck, Addison-Wesley, 1983.
- *Building Expert Systems*, F. Hayes-Roth, Waterman, and Lenat, eds., Addison-Wesley, 1983.
- *System Aids in Constructing Consultation Programs: EMYCIN*, van Melle, UMI Research Press, 1982.
- *Knowledge-Based Systems in Artificial Intelligence: AM and TEIRESIAS*, Davis and Lenat, McGraw-Hill, 1982.
- *The Handbook of Artificial Intelligence*, Volume I, Barr and Feigenbaum, eds., 1981; Volume II, Barr and Feigenbaum, eds., 1982; Volume III, Cohen and Feigenbaum, eds., 1982; Kaufmann.
- *Applications of Artificial Intelligence for Organic Chemistry: The DENDRAL Project*, Lindsay, Buchanan, Feigenbaum, and Lederberg, McGraw-Hill, 1980.

Our laboratory has pioneered in the development and application of AI methods to produce high-performance knowledge-based programs. Programs have been developed in such diverse fields as analytical chemistry (DENDRAL), infectious disease diagnosis and treatment (MYCIN), cancer chemotherapy management (ONCOCIN), pulmonary function evaluation (PUFF), VLSI design (KBVLSI/PALLADIO), molecular biology (MOLGEN), and parallel machine architecture simulation (CARE). Some of our systems and tools (e.g., UNITS, EMYCIN, and AGE) are now also being adapted for commercial development and use in the AI industry.

Following our lead in work on biomedical applications of AI and the development of the SUMEX-AIM computing resource, a nationally recognized community of academic projects on AI in medicine has grown up.

Central to all KSL research are our faculty, staff, and students. These people have been recognized internationally for the quality of their work and for their continuing contributions to the field. KSL members participate extensively in professional organizations, government advisory committees, and journal editorial boards. They have held managerial posts and conference chairmanships in both the American Association for Artificial Intelligence (AAAI) and the International Joint Conference on Artificial Intelligence (IJCAI).

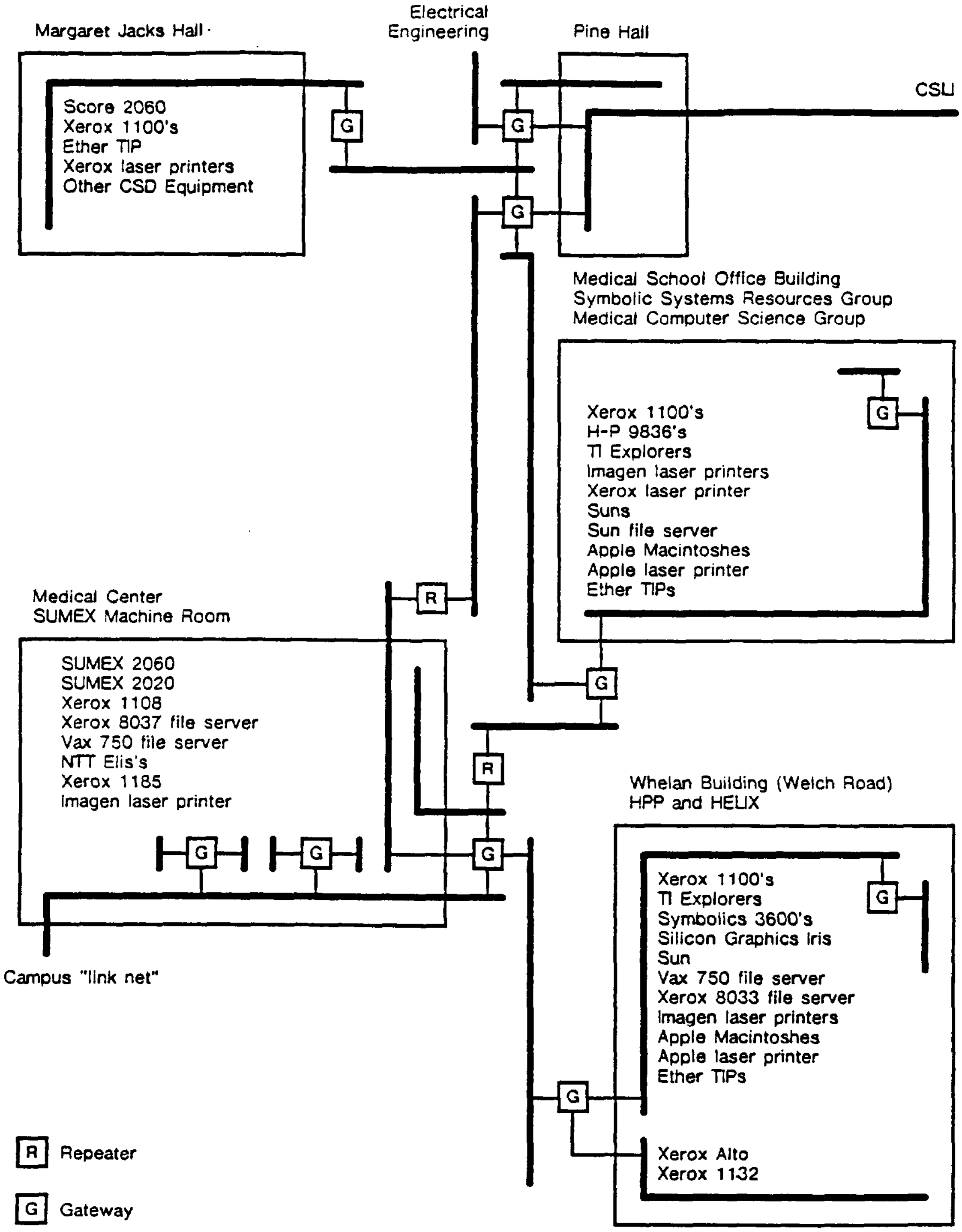
* Copies of individual KSL publications may be obtained through the Stanford Department of Computer Science publications office. The full collection of KSL reports is being published in microfiche by COMTEX Scientific Corporation.

Several KSL faculty and former students have received significant honors. In 1976, Ted Shortliffe received the Association of Computing Machinery Grace Murray Hopper award. In 1977, Doug Lenat was given the IJCAI Computers and Thought award, and in 1978, Ed Feigenbaum received the National Computer Conference Most Outstanding Technical Contribution award. In 1979 and 1981, Ted Shortliffe's book *Computer-Based Medical Consultation: MYCIN* was identified as the most frequently cited work in the IJCAI proceedings. In 1982, Doug Lenat won the Tioga prize for the best AAAI conference paper while Mike Genesereth received honorable mention. In 1983, Ted Shortliffe was named a Kaiser Foundation faculty scholar, and Tom Mitchell received the IJCAI Computers and Thought award. In 1984, Ed Feigenbaum was elected a fellow of the American Association for the Advancement of Science (AAAS), and he and Ted Shortliffe were elected fellows of the American College of Medical Informatics. In 1986, Ed Feigenbaum was elected to the National Academy of Engineering and in 1987, Ted Shortliffe was elected to the Institute of Medicine of the National Academy of Sciences.

## KSL Research Environment

**Funding**—The KSL is supported solely by sponsored research and gift funds. We have had funding from many sources, including DARPA, NIH/NLM, ONR, NSF, NASA, and private foundations and industry. Of these, DARPA and NIH have been the most substantial and long-standing sources of support. All, however, have made complementary contributions to establishing an effective overall research environment that fosters interchanges at the intellectual and software levels and that provides the necessary physical computing resources for our work.

**Computing Resources**—Under the Symbolic Systems Resources Group, the KSL develops and operates its own computing resources tailored to the needs of its individual research projects. Current computing resources are a networked mixture of mainframe host computers, Lisp workstations, and network utility servers, reflecting the evolving hardware technology available for AI research. Our mainframe host is currently a DEC 2060 running TOPS-20 (this is the core of the national SUMEX biomedical computing resource). Its network service functions will be replaced shortly by a SUN-4 system running UNIX. Its routine computing functions (electronic mail, text processing, and information retrieval) will be replaced by distributed user workstations. Our Lisp workstations include 35 Xerox 1100-series machines, 20 Texas Instruments Explorers, 6 Symbolics 3600-series machines, 3 SUN 3/75 workstations, and 5 Hewlett-Packard 9836 machines. We are in the process of acquiring a significant number of Apple Macintosh II workstations for routine computing support and many of these will also be configured to run Lisp programs. Network printing, file, gateway, and terminal interface services are provided by dedicated machines including 2 VAX 11/750's, a SUN 3/180, and numerous dedicated microprocessor systems. These facilities are integrated with other computer science resources at Stanford through an extensive Ethernet and to external resources through the ARPANET and TELENET. Funding for these resources comes principally from DARPA and NIH and hardware vendor gifts.



**SUMEX-AIM System and Local Area Network**

# Appendix B

# Lisp Performance Studies

## Performance of Two Common Lisp Programs on Various Workstation Systems

by Richard Acuff
Knowledge Systems Laboratory
Stanford University

*** DRAFT ***

### 1 - Introduction

In order to assist us in understanding performance of Lisp systems, we have undertaken an informal survey of Common Lisp environments using two KSL software packages. The data collection is close to complete but there has been very little data analysis. Thus the data is included here with very little in the way of observations or conclusions.

In this survey we have focused on execution speed which has long been a differentiator among computer systems. The first comparison of two systems solving the same problem (benchmarking) was probably done shortly after the creation of the second computer, and benchmarking has been a primary differentiator among computers systems ever since. However, execution speed benchmarks are only one aspect of the systems, especially Lisp systems. Issues like programming and usage environments, compatibility with other systems, ability to handle "large" problems, and cost must also be considered.

The test software we used was SOAR and the BB1 blackboard core. Both systems were chosen primarily because they are implemented in pure Common Lisp, making them extremely portable. Both are systems in daily use in the KSL and represent two distinct research directions. SOAR is a heuristic search based general problem solving architecture developed by Paul Rosenbloom and BB1 is a blackboard problem solving architecture developed by Barbara Hayes-Roth. Neither of these systems is an intensive user of numeric computation. These systems were initially developed in environments other than those tested and no attempt was made to optimize their performance for any of these tests.

All runs of SOAR were done solving an eight-puzzle problem in one of three modes:

1. Mode "1,3" just solves the problem.
2. Mode "1,1" solves the problem while learning how to better solve it (this mode takes the most time).
3. Mode "3,3" solves the problem after learning (this mode takes the least time).

SOAR's source code consisted of a single 280k character file, plus two small files containing the "rules" for the eight-puzzle problem: DEFAULT.SOAR at 24k characters and EIGHT.SOAR at 10k characters. The runs and compilation were done in separate instantiations of the Lisp environment.

All runs of BB1 went through three cycles of adding 10 items to the blackboard, accessing those 10 items, and then deleteing them. All references to BB1 in this document refer only to the "core" blackboard parts of the system and does not include any other layers of the problem solving architecture, or the user interface. The BB1 source code used for the testing is spread over 10 files containing 295k characters. Compilation, loading, and execution were all done in a single instantiation of the Lisp environment.

**2 - Systems Under Test**

The systems to be tested were chosen based on their availability to the testers as well as suspected potential usefulness in future programming efforts. Since we were interested in "real world" results, we ran the tests on each machine in what seemed to be its standard operating mode. In particular, if there are typically "background" activities going on during normal operation, then those were allowed to continue during the taking of these measurements. If code is typically executed from within an editor or other special context, then that was done. No special process priority altering or other attempt to optimize the execution was made unless noted in the description of the systems.

It is worth noting that on almost all of the systems tested, virtual memory paging was a neglibible part of the overall run time. Nor was it a significant factor during compilation.

In the following descriptions "Code" refers to a short name used to indicate the systems under test. Usually it is the model of the machine except where there is more than one Lisp for a machine (as in the case of the Sun 3/75) in which case a letter is prefixed to indicate the Lisp being used. "Timing Template" indicates how the information reported by the TIME function was recorded. "Elapsed" indicates the total elapsed time, "run" indicates CPU time used, "gc" indicates time spent in garbage collection, "user" and "system" distinguish between user mode and kernel mode time, and "paging" indicates time waiting for virtual memory disk operations. Though all of this information was recorded we have not reproduced it in this document.

Code: 3/260
Computer Type: Sun 3/260
Operating System: Sun OS 3.4
Lisp: Lucid 2.0
Disk Configuration: 280MB
Swapping Size: 60MB
Memory Configuration: 8MB
Display Configuration: Color in mono mode
Other Configuration:
Special Comments: used :EXPAND 130 :GROWTH-RATE 130
Timing Template: elapsed (user-run + system-run)

Code: 3/60
Computer Type: Sun 3/60
Operating System: Sun OS 3.4
Lisp: Lucid 2.1
Disk Configuration: SCSI 141MB
Swapping Size: unknown
Memory Configuration: 24MB
Display Configuration: Hi Res Color in mono mode
Other Configuration:
Special Comments:

# Appendix B

# Lisp Performance Studies

## Performance of Two Common Lisp Programs on Various Workstation Systems

by Richard Acuff
Knowledge Systems Laboratory
Stanford University

*** DRAFT ***

### 1 - Introduction

In order to assist us in understanding performance of Lisp systems, we have undertaken an informal survey of Common Lisp environments using two KSL software packages. The data collection is close to complete but there has been very little data analysis. Thus the data is included here with very little in the way of observations or conclusions.

In this survey we have focused on execution speed which has long been a differentiator among computer systems. The first comparison of two systems solving the same problem (benchmarking) was probably done shortly after the creation of the second computer, and benchmarking has been a primary differentiator among computers systems ever since. However, execution speed benchmarks are only one aspect of the systems, especially Lisp systems. Issues like programming and usage environments, compatibility with other systems, ability to handle "large" problems, and cost must also be considered.

The test software we used was SOAR and the BB1 blackboard core. Both systems were chosen primarily because they are implemented in pure Common Lisp, making them extremely portable. Both are systems in daily use in the KSL and represent two distinct research directions. SOAR is a heuristic search based general problem solving architecture developed by Paul Rosenbloom and BB1 is a blackboard problem solving architecture developed by Barbara Hayes-Roth. Neither of these systems is an intensive user of numeric computation. These systems were initially developed in environments other than those tested and no attempt was made to optimize their performance for any of these tests.

All runs of SOAR were done solving an eight-puzzle problem in one of three modes:

1. Mode "1,3" just solves the problem.
2. Mode "1,1" solves the problem while learning how to better solve it (this mode takes the most time).
3. Mode "3,3" solves the problem after learning (this mode takes the least time).

SOAR's source code consisted of a single 280k character file, plus two small files containing the "rules" for the eight-puzzle problem: DEFAULT.SOAR at 24k characters and EIGHT.SOAR at 10k characters. The runs and compilation were done in separate instantiations of the Lisp environment.

All runs of BB1 went through three cycles of adding 10 items to the blackboard, accessing those 10 items, and then deleteing them. All references to BB1 in this document refer only to the "core" blackboard parts of the system and does not include any other layers of the problem solving architecture, or the user interface. The BB1 source code used for the testing is spread over 10 files containing 295k characters. Compilation, loading, and execution were all done in a single instantiation of the Lisp environment.

## 2 - Systems Under Test

The systems to be tested were chosen based on their availability to the testers as well as suspected potential usefulness in future programming efforts. Since we were interested in "real world" results, we ran the tests on each machine in what seemed to be its standard operating mode. In particular, if there are typically "background" activities going on during normal operation, then those were allowed to continue during the taking of these measurements. If code is typically executed from within an editor or other special context, then that was done. No special process priority altering or other attempt to optimize the execution was made unless noted in the description of the systems.

It is worth noting that on almost all of the systems tested, virtual memory paging was a neglibible part of the overall run time. Nor was it a significant factor during compilation.

In the following descriptions "Code" refers to a short name used to indicate the systems under test. Usually it is the model of the machine except where there is more than one Lisp for a machine (as in the case of the Sun 3/75) in which case a letter is prefixed to indicate the Lisp being used. "Timing Template" indicates how the information reported by the TIME function was recorded. "Elapsed" indicates the total elapsed time, "run" indicates CPU time used, "gc" indicates time spent in garbage collection, "user" and "system" distinguish between user mode and kernel mode time, and "paging" indicates time waiting for virtual memory disk operations. Though all of this information was recorded we have not reproduced it in this document.

Code: 3/260
Computer Type: Sun 3/260
Operating System: Sun OS 3.4
Lisp: Lucid 2.0
Disk Configuration: 280MB
Swapping Size: 60MB
Memory Configuration: 8MB
Display Configuration: Color in mono mode
Other Configuration:
Special Comments: used :EXPAND 130 :GROWTH-RATE 130
Timing Template: elapsed (user-run + system-run)

Code: 3/60
Computer Type: Sun 3/60
Operating System: Sun OS 3.4
Lisp: Lucid 2.1
Disk Configuration: SCSI 141MB
Swapping Size: unknown
Memory Configuration: 24MB
Display Configuration: Hi Res Color in mono mode
Other Configuration:
Special Comments:

Timing Template: elapsed (user-run + system-run)

Code: 386
Computer Type: Compaq 386 (20Mhz 386)
Operating System: 386/IX 5.3 rev level 1.01 (unix)
Lisp: Lucid 2.0
Disk Configuration: 134MB ESDI
Swapping Size: unknown
Memory Configuration: 10MB; 32kB 20ns cache
Display Configuration: terminal
Other Configuration: none
Special Comments: none
Timing Template: elapsed (run)

Code: 386T
Computer Type: Compaq 386 portable (Toaster)
Operating System: 386/IX 5.3 rev level 1.01 (unix)
Lisp: Lucid 2.0
Disk Configuration: 40MB
Swapping Size: unknown
Memory Configuration: 10MB; no cache
Display Configuration: tiny LCD
Other Configuration: tiny display
Special Comments: portable versio of "386" above
Timing Template: elapsed (run)

Code: 4/260
Computer Type: Sun 4/280
Operating System: SunOS 3.2 Gamma
Lisp: Lucid 2.1
Disk Configuration: unknown
Swapping Size: unknown
Memory Configuration: 32MB
Display Configuration: Hi Res color in mono
Other Configuration:
Special Comments: used :EXPAND 130 :GROWTH-RATE 130
Timing Template: elapsed (user-run + system-run)

Code: 4/280
Computer Type: Sun 4/280
Operating System: SunOS 3.2 Gamma
Lisp: Lucid 2.1 beta
Disk Configuration: 417 (Eagle)
Swapping Size: 60MB
Memory Configuration: 8MB
Display Configuration: Hi Res mono
Other Configuration:
Special Comments:
Timing Template: elapsed (user-run + system-run)

Code: DEC-II
Computer Type: DEC MicroVax II/GPX
Operating System: VMS
Lisp: VaxLisp
Disk Configuration: 2 x 159MB
Swapping Size: 3k pg page, 8k pg swap

Memory Configuration: 16MB
Display Configuration: GPX
Other Configuration:
Special Comments:
Timing Template: elapsed - gc-elapsed (run - gc-run)

Code: DEC-III
Computer Tyoe: DEC MicroVax III (3500)
Operating System: VMS
Lisp: VaxLisp
Disk Configuration: (RD53)
Swapping Size: unkown
Memory Configuration: 16MB
Display Configuration:
Other Configuration:
Special Comments:
Timing Template: elapsed - gc-elapsed (run - gc-run)

Code: E-3/75
Computer Type: Sun 3/75
Operating System: SunOS 3.1
Lisp: Franz Extended Common Lisp 2.0
Disk Configuration: 70MB SCSI
Swapping Size: 50MB local
Memory Configuration: 28MB
Display Configuration: standard resolution mono
Other Configuration: Files on Sun 3/180 NFS server
Special Comments: Under suntools
Timing Template: elapsed (run + gc)

Code: EXP1
Computer Type: Texas Instruments Explorer I
Operating System: Explorer Lisp Release 3.0+
Lisp: Explorer Lisp Release 3.0+
Disk Configuration: 2 x 140MB SCSI
Swapping Size: 80MB
Memory Configuration: 8MB
Display Configuration: 1024 x 768 mono
Other Configuration:
Special Comments: TGC (incremental generation scavenging GC) on unless otherwise noted
Timing Template: elapsed - paging

Code: EXP2
Computer Type: Texas Instruments Explorer II
Operating System: Explorer Lisp Release 3.0+
Lisp: Explorer Lisp Release 3.0+
Disk Configuration: 2 x 140MB SCSI
Swapping Size: 80MB
Memory Configuration: 16MB
Display Configuration: 1024 x 768 mono
Other Configuration:
Special Comments: TGC (incremental generation scavenging GC) on unless otherwise noted
Timing Template: elapsed - paging

Code: HP
Computer Type: Hewlett Packard 9000/350
Operating System: Unix
Lisp: HP Lisp 1.0
Disk Configuration: 130MB (7958)
Swapping Size: unknown
Memory Configuration: 16MB
Display Configuration: color
Other Configuration: under gnuemacs
Special Comments:
Timing Template: elapsed - run

Code: K-3/75
Computer Type: Sun 3/75
Operating System: SunOS 3.1
Lisp: Kyoto Common Lisp "September 16, 1986"
Disk Configuration: 70MB SCSI
Swapping Size: 50MB local
Memory Configuration: 28MB
Display Configuration: standard resolution mono
Other Configuration: Files on Sun 3/180 NFS server
Special Comments: Under suntools
Timing Template: elapsed - run

Code: L-3/75
Computer Type: Sun 3/75
Operating System: SunOS 3.1
Lisp: Lucid 2.0
Disk Configuration: 70MB SCSI
Swapping Size: 50MB local
Memory Configuration: 28MB
Display Configuration: standard resolution mono
Other Configuration: Files on Sun 3/180 NFS server
Special Comments: used :EXPAND 90 :GROWTH-RATE 90
Timing Template: elapsed (user-run + system-run)

Code: mX
Computer Type: Texas Instruments microExplorer
Operating System: Explorer Lisp 4.0 beta
Lisp: Explorer Lisp 4.0 beta
Disk Configuration: 100MB Rodime
Swapping Size: 60MB
Memory Configuration: 12MB mX processor; 2MB Mac II
Display Configuration: 19" (1024 x 768) Moniterm Viking
Other Configuration: Apple EtherTalk
Special Comments:
Timing Template: elapsed - paging

Code: RT
Computer Type: IBM RT/APC
Operating System: AIX 2.1.2 (unix)
Lisp: 2.0.5 (Lucid 1.01)
Disk Configuration: "Fast" EESDI controller; 3 x 70MB
Swapping Size: 80k x 512kB blocks (40,960MB)
Memory Configuration: 16MB of "fast" memory
Display Configuration: Moniterm 1024 x 768 mono

Other Configuration: AFT floating point unit; GSL windows
Special Comments:. Used :EXPAND 69 to get 6MB semispace; This should be the fastest RT version now available
Timing Template: elapsed (user-run + system-run)

Code: Sym
Computer Type: Symbolics 3645
Operating System: Symbolics Release 6.1
Lisp: Symbolics Release 6.1
Disk Configuration: 368MB
Swapping Size: 200MB
Memory Configuration:
Display Configuration: 8MB
Other Configuration: FPA, no color
Special Comments: EGC on
Timing Template: elapsed - paging

Code: XCL
Computer Type: Xerox 1186
Operating System: Xerox Lisp, Lyric release
Lisp: Xerox Lisp, Lyric release
Disk Configuration: 40MB
Swapping Size: 16MB
Memory Configuration: 3.5MB
Display Configuration: 19" mono
Other Configuration:
Special Comments:
Timing Template: elapsed - gc - paging

### 3 - Compilation and Execution

For both BB1 and SOAR the time taken to compile the system and make a standard run was measured. Here are those results (all numbers are seconds):

| | BB1 | | SOAR | |
|---|---|---|---|---|
| Code | Compile | Run | Compile | Run |
| 3/260 | 540 | 62 | 687 | 171 |
| 3/60 | 551 | 73 | 569 | 218 |
| 386 | 355 | 47 | 386 | 142 |
| 386T | 416 | 54 | 479 | 175 |
| 4/260 | 324 | 56 | 307 | 70 |
| 4/280 | 482 | 34 | 523 | 105 |
| DEC-II | 1774 | 207 | 1227 | 1908 |
| DEC-III | 633 | 63 | 423 | 476 |
| E-3/75 | 444 | 211 | 450 | 500 |
| Exp1 | 327 | 87 | 520 | 400 |
| Exp2 | 96 | 29 | 162 | 146 |
| HP | 235 | 115 | 237 | 229 |
| L-3/75 | 919 | 90 | 1365 | 756 |
| K-3/75 | 1234 | 96 | 1040 | 297 |
| MacII | 349 | 254 | Not available[1] | |
| mX | 242 | 31 | 242 | 219 |
| RT | 586 | 75 | 574 | 206 |
| Sym | 257 | 111 | 252 | 210 |
| XCL | 1927 | 559 | 1800 | 1613 |

[1] We were not able to get SOAR to run properly in Allegro Common Lisp 1.0 or 1.1.

For convenience of viewing, these numbers are graphed in Figures 20 and 21. The system types are sorted into order of best run time for SOAR and BB1 separately to facilitate comparative observation.

### 4 - Effect of Compiler Optimize Settings on BB1

We were also interested in the effect of two of Common Lisp's compiler optimizer settings, SPEED and SAFETY. These switches have four settings, 0 through 3, with 0 being the highest priority. Thus settings of SAFETY 0 and SPEED 3 should allow the compiler to produce the fastest code, while SAFETY 3 and SPEED 0 would result in conservative code, perhaps with more type checking, etc. We comipiled and ran BB1 with 4 settings of these switches:

1. The system default
2. (PROCLAIM '(OPTIMIZE (SAFETY 0) (SPEED 3)))
3. (PROCLAIM '(OPTIMIZE (SAFETY 3) (SPEED 0)))
4. (PROCLAIM '(OPTIMIZE (SAFETY 2) (SPEED 3)))

Figures 22 and 23 show the effect of these settings on the compilation and run times of the BB1 test. Here are the numbers:

| | Default | | Safe 0, | Spd 3 | Safe 3, | Spd 0 | Safe 2, | Spd 3 |
|---|---|---|---|---|---|---|---|---|
| Code | Comp | Run | Comp | Run | Comp | Run | Comp | Run |
| 3/260 | 540 | 62 | 524 | 62 | 532 | 69 | 527 | 62 |
| 3/60 | 551 | 73 | 537 | 72 | 444 | 76 | 540 | 72 |
| 386 | 355 | 47 | 332 | 47 | 271 | 52 | 339 | 47 |
| 386T | 416 | 54 | 408 | 54 | 341 | 60 | 410 | 54 |
| 4/260 | 324 | 56 | 318 | 46 | 229 | 47 | 309 | 46 |
| 4/280 | 482 | 34 | 483 | 34 | 385 | 48 | 492 | 34 |
| DEC-II | 1774 | 207 | 1987 | 206 | 2245 | 231 | 3094 | 236 |
| DEC-III | 633 | 63 | 635 | 60 | 732 | 71 | 990 | 70 |
| E-3/75 | 444 | 211 | 403 | 215 | 469 | 206 | 443 | 206 |
| Exp1 | 327 | 87 | 318 | 87 | 314 | 90 | 357 | 83 |
| Exp2 | 96 | 29 | 117 | 25 | 111 | 28 | 121 | 26 |
| HP | 235 | 115 | 250 | 113 | 235 | 141 | 247 | 118 |
| K-3/75 | 1234 | 96 | 1815 | 165 | 1379 | 147 | 1171 | 88 |
| L-3/75 | 919 | 90 | 1056 | 90 | 1054 | 127 | 910 | 90 |
| MacII | 349 | 254 | 365 | 258 | 354 | 261 | 363 | 259 |
| mX | 242 | 34 | 249 | 28 | 232 | 32 | 239 | 35 |
| RT | 586 | 75 | 587 | 76 | 508 | 77 | 595 | 75 |
| Sym | 257 | 111 | 281 | 109 | 256 | 110 | 262 | 111 |
| XCL | 1927 | 559 | 2022 | 543 | 2230 | 559 | 2020 | 556 |

### 5 - Effect of Output Reduction on SOAR

We had previously noted that some systems were able to run the eight-puzzle benchmark much faster when the volumous typeout produced was reduced. Figure 24 shows the difference between run times of the 1,3 mode solution of the eight puzzle with full tracing versus no tracing. The numerical data follows:

| Code | Normal | Reduced |
|---|---|---|
| 3/260 | 49 | 33 |
| 3/60 | 66 | 38 |
| 386 | 41 | 27 |
| 386T | 52 | 31 |
| 4/260 | 23 | 13 |
| 4/280 | 35 | 15 |
| DEC-II | 351 | 283 |
| DEC-III | 95 | 76 |
| E-3/75 | 124 | 109 |
| Exp1 | 90 | 63 |
| Exp2 | 44 | 21 |
| HP | 61 | 51 |
| K-3/75 | 186 | 136 |
| L-3/75 | 82 | 67 |
| mX | 48 | 38 |
| RT | 61 | 36 |
| Sym | 55 | 40 |
| XCL | 473 | 390 |

## 6 - Future Work

Now that this data is collected it is our intention to write it up in a technical report, including comparisons with the Gabriel Benchmarks and remarks on certain surprising facts that this work has turned up. This report will be made widely available to the AIM community to assist future descisions in the use of Lisp systems.

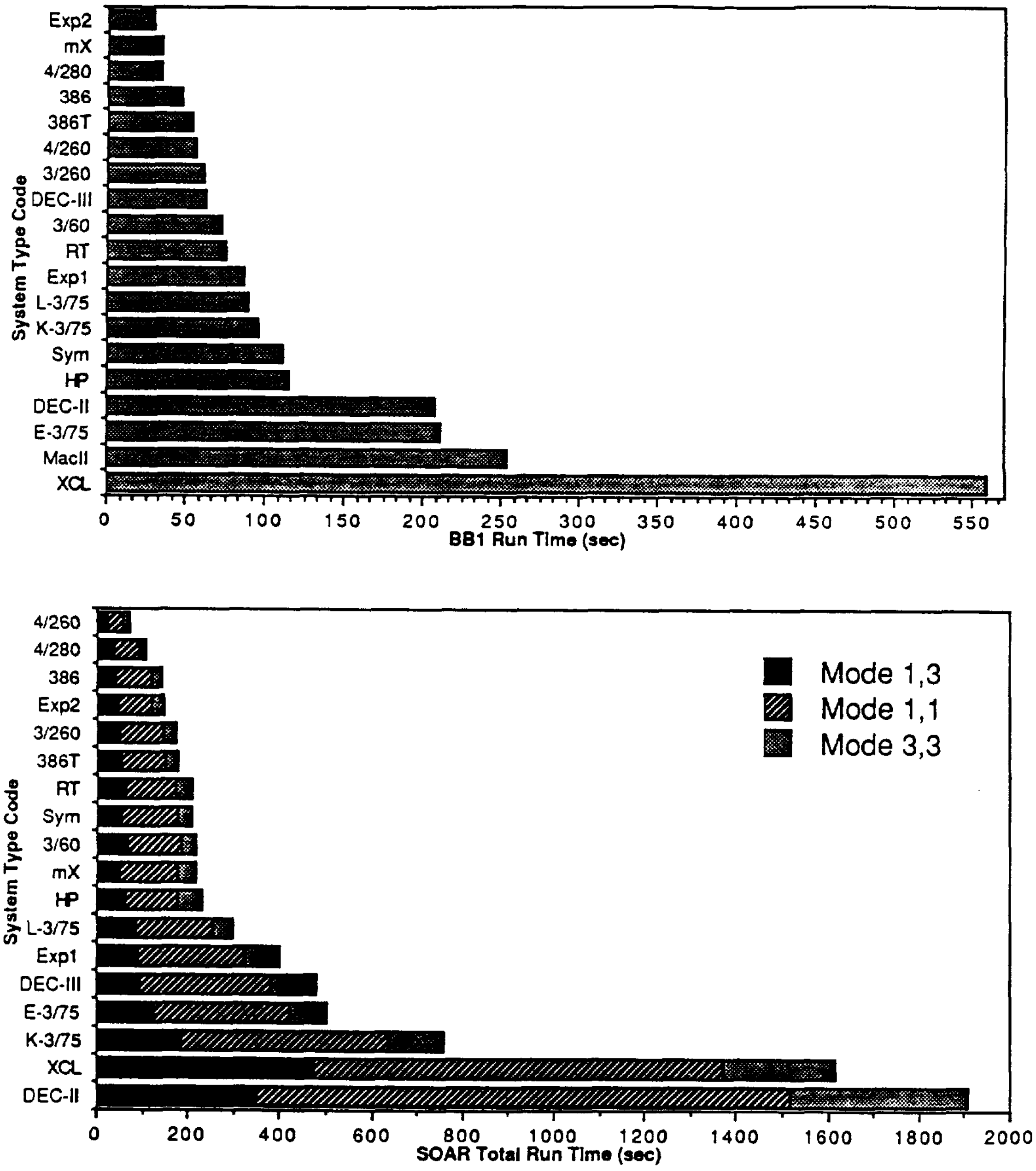


**Figure 20:** Run Times for BB1 and SOAR

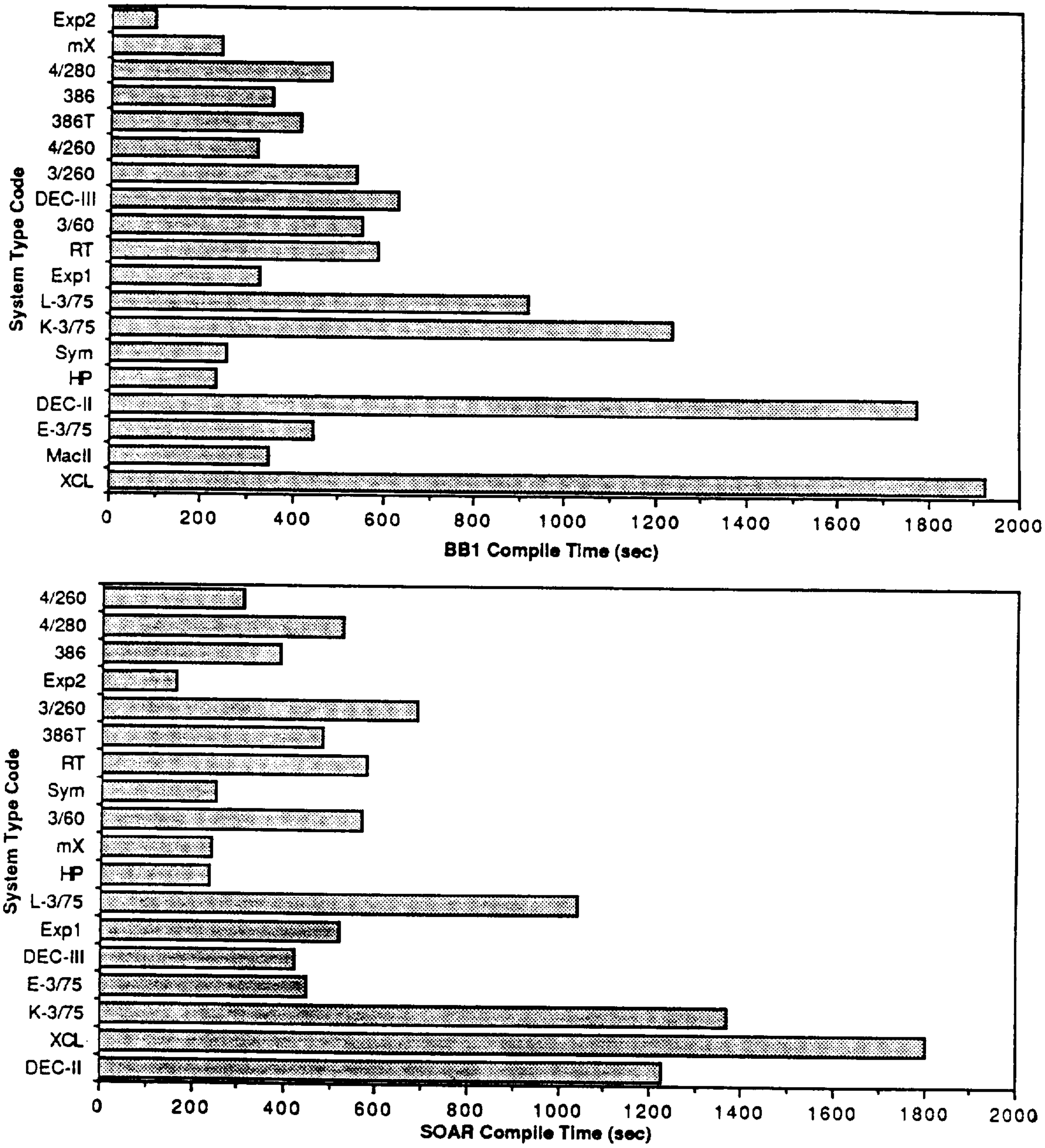


**Figure 21:** Compilation Times for BB1 and SOAR

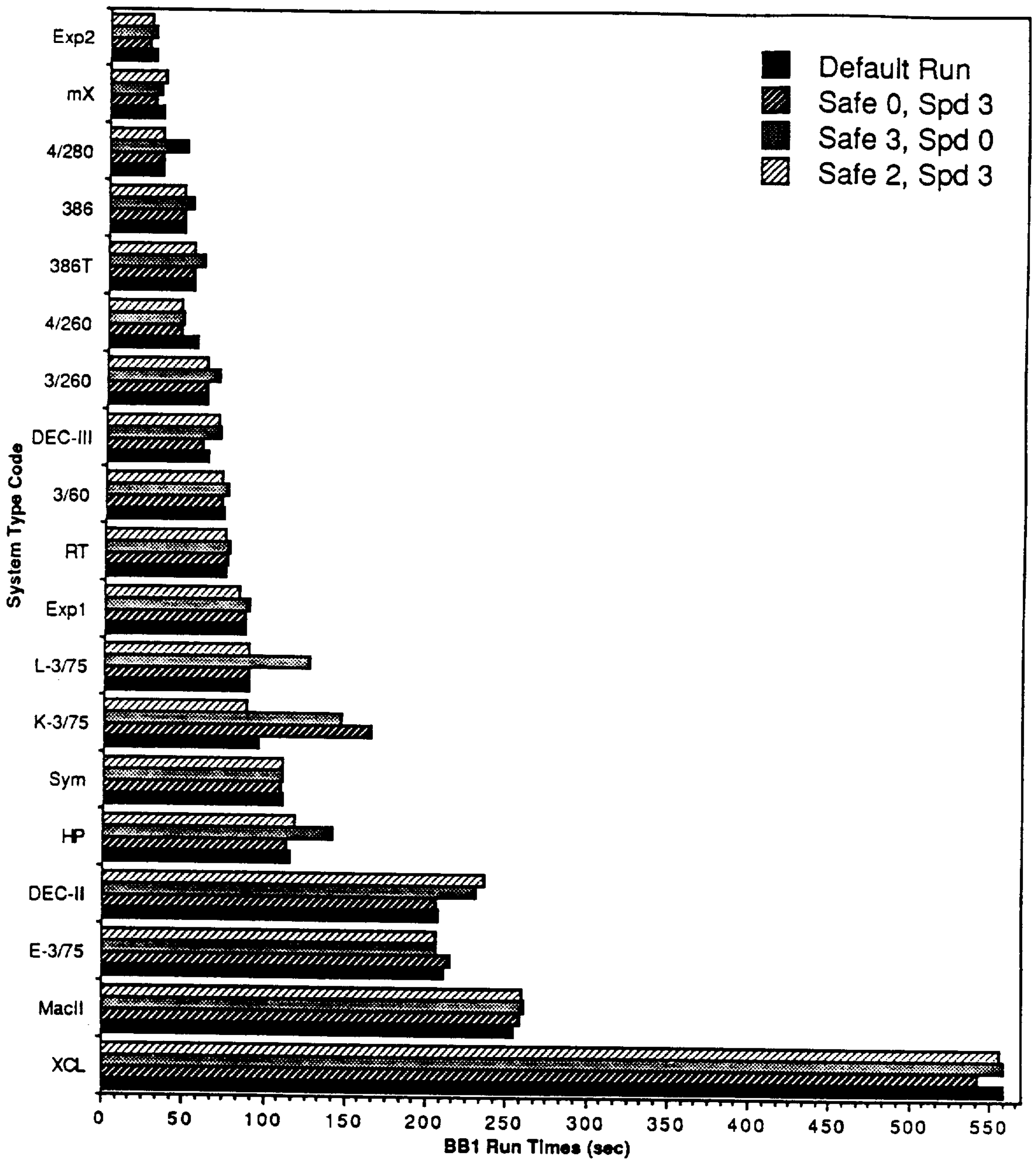


Figure 22: Run Times for BB1 under Various Compiler Settings

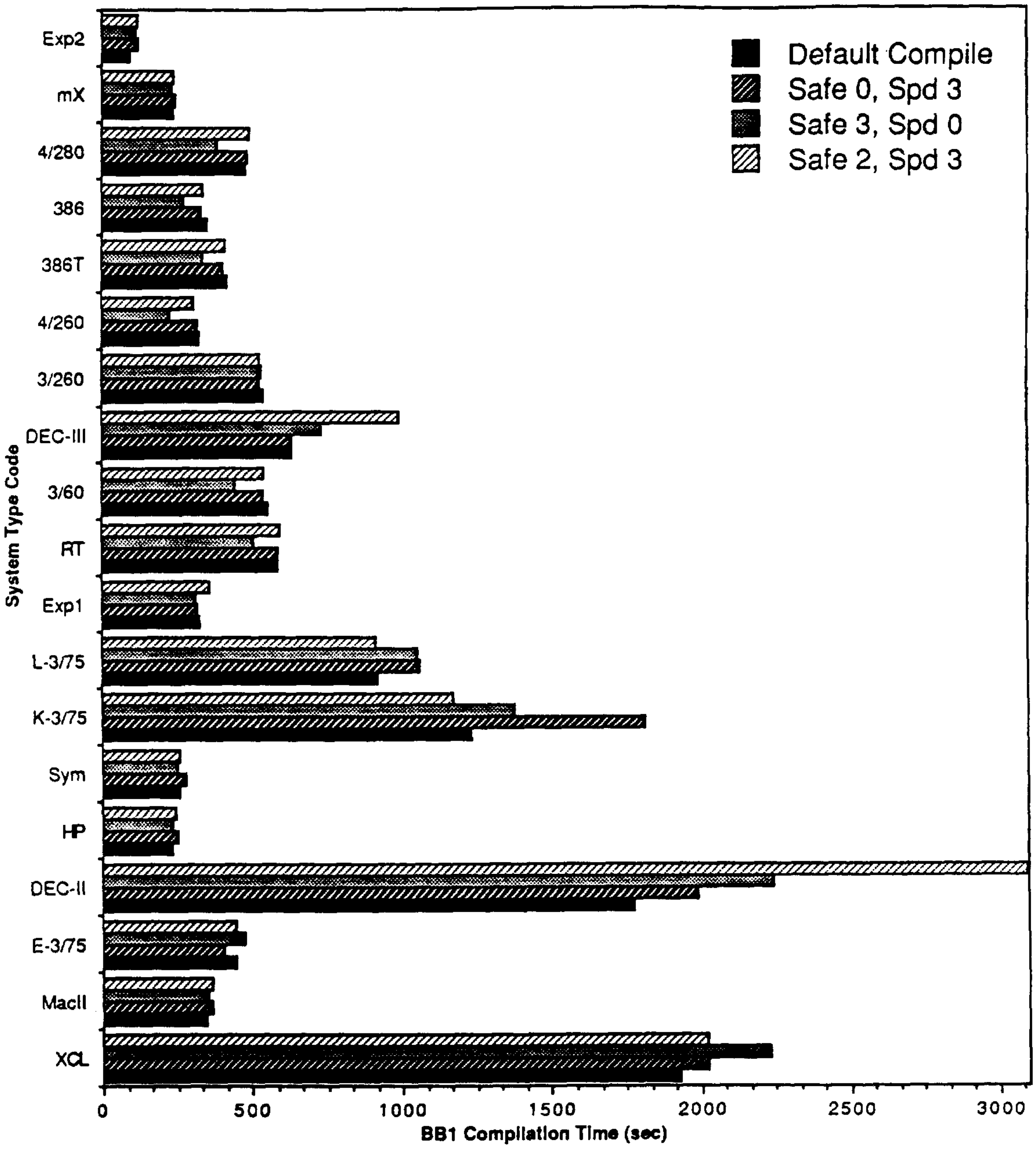


**Figure 23:** Compilation Times for BB1 under Various Compiler Settings

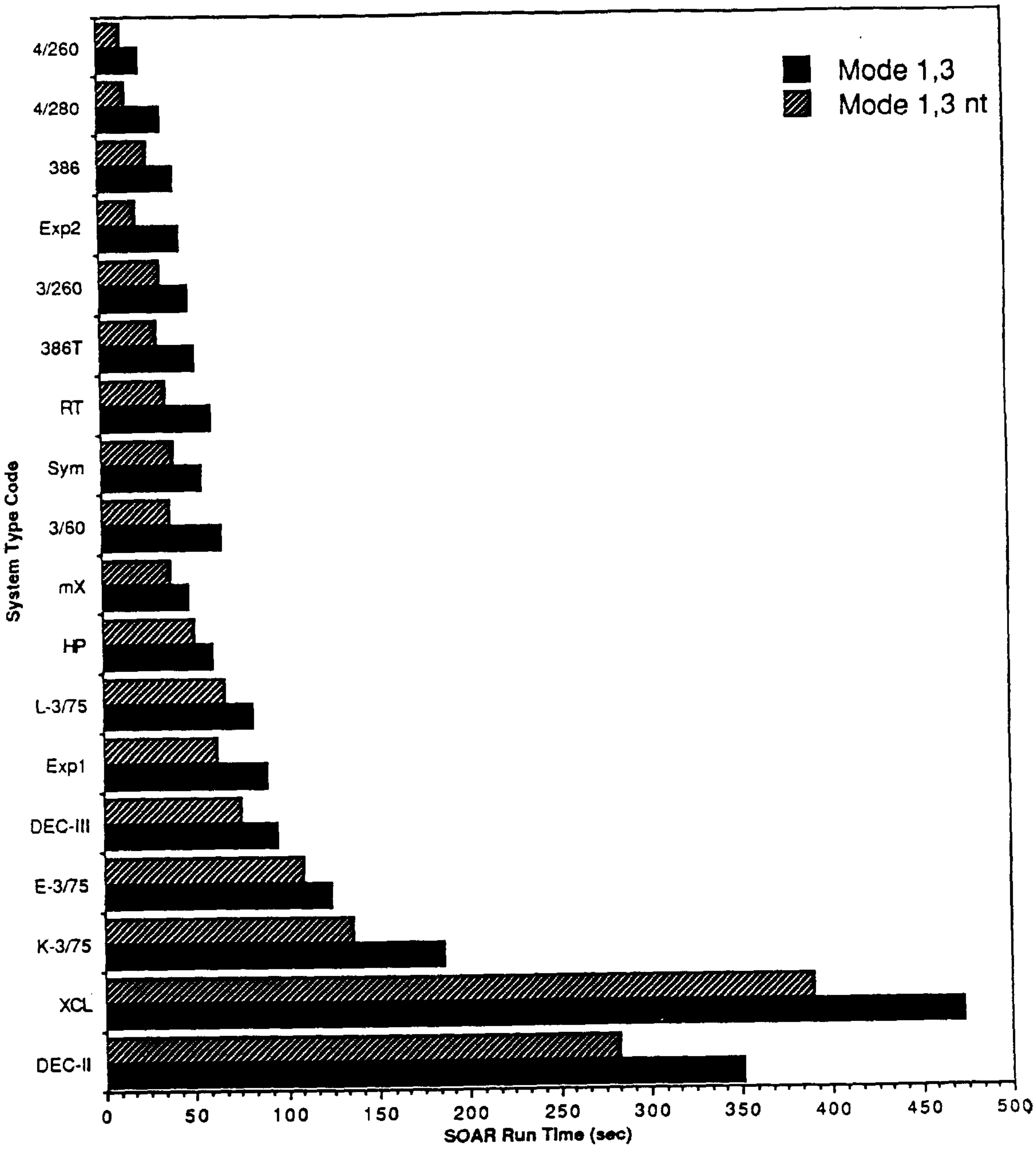


Figure 24: Differences Between Normal and Reduced Ouput SOAR Run

## Appendix C
## Lisp Environment Specification

# KSL Lisp Environment Requirements

# *** DRAFT ***

Richard Acuff

Knowledge Systems Laboratory

Computer Science Department, Stanford University

May 11, 1988

# 1. Introduction

The Knowledge Systems Laboratory (KSL) is an artificial intelligence research laboratory at Stanford University that has been developing systems and tools in Lisp environments for more than 20 years. These systems have been implemented in the InterLisp, MACLisp, ZetaLisp, and more recently, CommonLisp dialects predominantly. Beginning in the early 1980's, our work moved from mainframe Lisp environments to workstation environments for many reasons, principally involving powerful tools for system development and debugging and graphical interfaces. Commercial versions of these tools, that evolved over many years in the Xerox D-Machine, Symbolics 36xx, LMI, and TI Explorer systems, have become an indispensible part of our work environment. Newer Lisp systems for workstations not specifically developed for Lisp have lacked many important features of these environments. This document attempts to summarize the key features of the Lisp machine environments that would be needed in "stock" machine implementations in order to make them attractive in a development setting.

There are several overall points to be emphasized about this write-up:

1 ) These requirements represent a snapshot of the tools and technology available on today's machines. AI has historically and will continue to ride the crest of the wave of new computing technologies for the forseeable future, which enable ever more complex systems. Thus, these are not static requirements and we expect to be able to take advantage of the future improvements in hardware, graphics, and software as they are generated by computer science research and industry.

2) It has been hard to describe concisely many aspects of the Lisp environments because they involve visual interactions and the "feel" of the way systems are organized and interconnected. The write-up assumes a general familiarity and experience with the Lisp environments Xerox, Symbolics, TI Explorer systems.

3 ) We have tried to sort out the key features of current systems that are important to our research work. Except where explicitly stated, everything in this document describes this "core" of functionality. Some items are clearly more important than others, but all represent needs that really guide our decisions about which new systems can be broadly used in the KSL.

4 ) The discussion is organized according to a "layered" view of Lisp environments shown in Figure 1, beginning with the upper levels. This organization is a conceptual framework within which to describe the various parts of the environment but may not correspond in full detail to the way system modules are actually organized. While most of the discussion focuses on the higher layers in this diagram, unavoidably some issues involving lower level or more general issues such as address space, system speed, and graphics facilities have to be mentioned.

5 ) While this description is based on what we know, use, and understand today, we have attempted to allow for innovation by describing the functionality that we require in a fairly abstract way wherever possible, rather than specifying, for example, the "Symbolics XYZ feature". This may result in

some ambiguities that will need to be addressed by appropriate discussion and iteration.

6 ) We have two overriding goals in adopting future computing environments (which may seem to be or actually are in some conflict). We want the most powerful development environments we can get to facilitate the building of complex AI programs. But at the same time, we want to be able to share (import and export) research results and tools with colleagues in other labs and so must maximize the portability of code among systems. We believe that these goals can be approached jointly by the establishment and careful adherence to standards where possible, while continuing systems development where necessary.

7 ) Because of the evolving nature of these research environments, no vendor's system can ever be "finished". While we expect reasonably professional standards of robustness and reliability in the systems we use, we also expect to have special needs and to work closely with the vendors of products we use to adapt, extend, and debug the environment and tools. Our experience has been that in order to do this effectively, it is essential that we have broad access to system source codes.

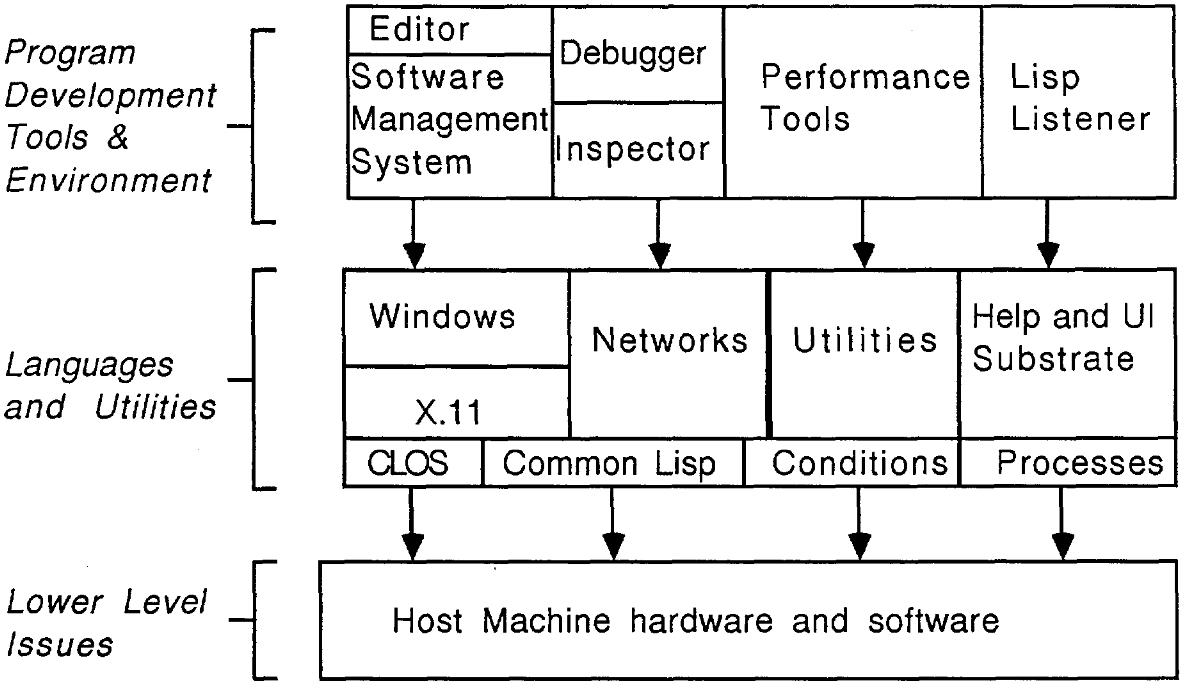


**Figure 1**

## 2. Program Development Tools & Environment

The quality of the development tools and environment is what has been the primary strength of Lisp machines, allowing rapid design, implementation, and debugging of complex programs. We believe the key to good development tools is integration, both in terms of consistency of interface, and in the ability to move seamlessly from tool to tool, carrying along appropriate data and state information. These qualities must be manifest in any KSL research computing system.

## 2.1. Editor

The hands of the development environment is the editor. There has been a great deal of experimentation with various styles of editing, most significantly text-based, as in Zmacs on the TI Explorer and Symbolics machines, versus structure-based, as in Xerox Lisp. We are inclined to believe that an editor rooted in a text-based approach but having understanding of the structural content of code being edited is the best approach as it allows full base-level generality for dealing with all kinds of text but, if well implemented, can be specialized for various types of editing. Given this, we feel the editor in the Lisp systems should have the following features:

- since it is common to build tools that utilize editing it is important to have a complete programatic interface to the editor
- able to use Emacs-like commands for hands-on-keyboard control as in Zmacs
- also uses pointer for moving editing focus, selection, some command selection, etc.
- fully extensible in Lisp
- uses the pretty printer described above to allow code reformatting (eg. for narrower/wider windows)
- minimally includes some source libraries to be used as examples
- supports keyboard macros
- integrated with Lisp such that edit definition, incremental compilation, documentation string viewing, argument list viewing, macro expansion, evaluation, etc. are available easily from the editing environment
- knows lisp syntax such that users can manipulate Lisp expressions (eg. move forward on s-expression, select an s-expression, etc.)
- has a complete edit definition facility such that the source of a DEFSTRUCT, DEFUN, DEFCLASS, or other definition of symbol will be automatically loaded if available without the need for explicit cross referencing
- allows user-defined modes for non-Lisp
- if text can be selected and operated on the selected text should be highlighted as in region marking in Zmacs on the TI Explorer
- if the matching grouping character is visible when the editing focus is on a grouping character, it should be indicated; a "grouping character" is a parentheses, curly brace, square brace, angle bracket, double quote, or other user-specified character; a "match" is found when a symmetric character is found in a syntactically legal place (ie. not in a different context than the focus character, such as in a comment or literal string when the focus

character is in code), thus requiring that the editor have "understanding" of the syntax being used

- allows multi-fonting and font shift stripping (eg., writes #2\a to files)

- has the ability to automatically place code into different fonts depending on context

- if the editor allows code with unbalanced parentheses to be entered, it should be possible to check the code for unbalanced parentheses, and such a check should be done when the code is saved in a file (eg. x-X Find Unbalanced Parentheses in TI and Symbolics' Zmacs)

- can be instantiated multiple times (multi-window and multi-process)

- completion of commands and file names

- hooks for buffer switching, buffer creation, mode changes

- per buffer/window editory control variable bindings

- ZetaLisp style attribute lists or some other mechanism for telling the editor what packages, fonts, base, etc. are used with data

### 2.2. Debugger

The interactive debugger is also a critical part of the Lisp environment. In many ways, it can be viewed as an extension of the inspector. It should have the following features:

- "Terminal" based and window based versions for times when the window system fails

- ability to force entry to debugger from keyboard, or abort execution of running program

- ability to see and modify arguments and locals of active stack frames and closures, as well as evaluating expressions in the lexical context of stack frames running compiled code

- ability to peruse the stack easily and quickly

- ability to return from or restart an arbitrary stack frame

- able to have multiple concurrent instantiations

- *must* be robust in the face of errors that occur during the operation of the debugger (eg. can't print some datum)

- arglists and docstrings must be around and accessible

- fast startup

One of the most difficult aspects of debugging is understanding where in a program the error is occurring. To assist with this, we require a good disassembler and some level of source code debugging. The disassembler must be able to give information about operations being performed including names of variables, indication of arguments being set up and functions being called, and current execution location.

The source code debugger should indicate either the current source form being executed in a given stack frame or the most recently exited form, and allow entering the editor on that source code. It is acceptable to have to compile code with a special flag on or with special declarations in effect in order to achieve source code debugging, and a moderate (approximately twofold at the most) performance penalty in code compiled with source code debugging in effect is acceptable. We have implemented such a system at the KSL by storing the program location counter (PC) before and after the execution of each form and using this to index back to the source code. It is *not* required that the source code debugger (SCD) work with all optimizations, but it should be possible to disable those optimizers to use the SCD and still run effectively.

### 2.3. Inspector

The backbone of the development environment is the Inspector. This tool must be quite flexible and user customizable. Given a datum the Inspector should display it in a window in such a way as to make the structure and contents quickly apparent to the user. For instance, a DEFSTRUCT structure might be displayed with a column with field names on the left and values on the right. It should further be possible to ask for alternative perspectives so that the user could view a list as a simple list of items, an ALIST, or a PLIST, and similarly for structures. The mouse should be used to traverse data structures by further inspecting. The display should not be strictly tabular to admit to nested data, graphs, etc.. The Inspector should also have the following attributes:

- there should be a well-defined protocol to allow instances to display themselves and have non-standard mouse sensitivity in the inspector

- able to work with all CL types, CLOS objects, compiled code, stack groups, and other objects that can be found in the system.

- startup quickly

- format only what is visible so that users don't have to wait for formatting of large data structures of which they wish to view only a small part

- allow fields to be modified easily

- handle circular data structures, preferably using the Common Lisp #1=(a b #1#) notation; (a b ...) is not acceptable.

- use multiple windows (one per datum) with a separate history window

- support concise and verbose modes so that, for instance, an instance being viewed as part of another structure could display itself briefly, but be more detailed when being viewed directly

### 2.4. Software Management Tools

An often neglected component of the development environment is a tool to manage software systems, keeping track of versions, patches, compilation and load dependencies, etc. This has frequently been handled in the past with simple command procedures and the file system. These primitive mechanisms fail to handle many cases that are becoming more and more important such as version checkpointing and multiple programmers, as well as needing version numbers in the host file system to support backup versions. We have explicitly not required version numbers on files, but only under the condition that the usefulness of the version numbers is addressed in the software management system (SMS). The SMS should have the following features:

- in the SMS, even more than usual, quick response and non-intrusive function are critical so that users aren't tempted to circumvent the system, thus ruining its integrity
- allow multiple versions of objects to be kept for both backup and for release cycling
- allow for patches to "released" systems
- allow partial or complete recompilation of systems, automatically taking care of dependencies
- allow transitive dependencies so that if system A depends on system B and system C depends on system A, manipulations of C cause both A and B to be affected appropriately
- support team programming via object or module "check out" (ie. only one programmer is able to write a module, and ideally audit trails are kept); the the smaller the module size, the less chance for two programmers requiring write access to a module at the same time
- can be either file or object based

## 2.5. Performance Monitoring and Analysis

An important aspect of writing software is the ability to find out where programs are spending their time so that tuning work can be applied appropriately to sluggish programs. Thus we require the following performance measurement facilities:

- stack sampling wherein a record of what functions are active on the stack is recorded at small intervals
- function entry counting
- accurate meters; microsecond precision desired

## 2.6. Lisp Listener

There must be a "listener" or "top-level" which is how the user interacts with the read-eval-print loop of Lisp. Along with the terminal oriented listener there should be a window based implementation (ideally based on the system editor) with the following features:

- a history mechanism allowing access to past typein and results (in at least a text form)
- editing ability, including using the pointer

# 3. Languages and Utilities

## 3.1. Windowing

Currently, the best way for a computer to present information to an interactive user seems to be via digital images presented on CRT displays. The display is divided into sub-displays called "windows" allowing various pieces of information to be presented at once. The programmatic and user interfaces to this mechanism is called the "window system".

It is very difficult to fully specify a window system complete, flexible, and efficient enough to be what everybody needs and will need. Therefore, the most important aspect of the desired window system is that it be able to evolve as more is learned about user interface and data presentation. Therefore the window system must be well layered and modular to support incremental mutation and experimentation.

In particular, we expect to see tools on top of an application toolbox, on top of window system primitives, on top of a window transport protocol (such as CLX with X.11), with the inter-layer communication passing through well-defined CLOS protocols such that new layers can be implemented with a minimum of trouble. In particular, we also expect to routinely use remote windowing capabilities, so it's very important that the windowing protocol be a standardized one accessible from many machines, such as X.11. Standards in other layers should be used as they become available and appropriate.

The layering approach also allow flexibility in display devices, and if well implemented, would allow easy redirection of output to a "display" device that happens to be a printer to give high-resolution hard copy, such as seen in Xerox Lisp's ImageOp facility, as well as color displays, higher or lower resolution displays, files, etc.

We expect people to be developing tools under at least two different windowing paradigms, including the "messy desk" metaphor which is characterized by many small (relative to the display size) windows each of which is an application or piece thereof, like a desk with many papers on it, and the "display swapping" metaphor in which there are a few applications, each of which typically takes the whole screen when active, though the individual applications usually have smaller "panes" in the display-filling "frame", and the user swaps which application is on the display via some keystroke sequence. There are arguments for both styles, and we would like our next generation system to support both styles to the extent possible. Generally, we feel the "messy desk" approach is the more general of the two, and the more sought after, and so should receive the most attention.

With that general framework, here are some specific requirements for the window system (Note: the "as in" comments below are intended to give examples of current systems with the type of functionality we wish to describe, and are not indended as strict specification of how the functionality should be implemented):

- since we can't predict the needs of our programmers in this time of rapidly changing user interface technology, we have to insist on access to the source code for at least the higher levels of the window system, and that the inter-layer protocols be well documented and flexible

- while we expect almost all interactions to be window-based, it will be necessary at times to access the Lisp from a non-windowing system or terminal so we need terminal-like interfacing to be available on at least a rudimentary basis making it possible, for instance, to check on a long-running program from a remote location over a crude link

- provide programmer specifiable on-screen mini-doc about available mouse actions, and use this facility in system tools
- there must be a way to reestablish connections to the window system from remote hosts without restarting Lisp so that if, say, a network connection fails the user can reconnect and reattach to the Lisp session if the operating system hasn't killed it
- horizontal and vertical scrolling based on customizable redisplay (wherein the user programs how to fill in the newly exposed window area) such as in Symbolics Lisp
- hierarchical (nested) window structuring as in TI Explorer Lisp
- high tolerance for "logical" errors, so that minor operational errors, such as incorrectly reshaping a window, don't cause the window system to crash
- non-restrictive parameters such the maximum number of windows, depth of the window hierarchy, size of a window, etc.
- possible to write to non-exposed windows as in the Apple Macintosh
- a window title mechanism (with option of attaching mouse handlers to titles) as in TI Explorer Lisp
- customizable scroll bars (eg. size, location, color, pattern, pop-up, mouse-capturing, mouse-click actions, extra "buttons", etc.)
- constraint frames that allow automatic configuration of inferior windows when the superior is altered as in TI Explorer Lisp
- fast opening windows, with no more than fraction of a second delay for typical windows in typical circumstances
- scrolling and character drawing rate of at least 1,500 char/sec in a full-screen window
- operations that work in color and B/W
- optional color
- use n-dimensional abstract positions (not X and Y coordinates), even if the window system only uses 2-d points in order to allow for future display devices that may well be able to deal in 3-d
- documented font formats so users may define new fonts
- ability to use any font, any time without using a "font map" (requiring separate operations to ensure proper baseline calculations is ok)

- customizable, titled pop-up menu styles with 2 standard styles: roll-out (the menu stays in place with no buttons down until an item is selected or the mouse is moved away from the menu as in TI Explorer Lisp) and button-up (the menu stays up while a mouse button is held down, selecting the item the mouse is over when the button is released or none if the mouse is outside the menu as in Xerox Lisp); "pull down" menus should be implementable
- a "snapshot" facility to allow a section of the screen image to be recorded in another window for later viewing, printing, or saving on a file in a published raster format
- ability to shrink application windows into smaller icons as in Xerox Lisp
- customizable event distribution (mouse clicks, etc.)
- user-extensible pop-up windows used for entering data ("dialog box") such as seen on the Apple Macintosh dialog boxes or the TI Explorer Choose Variable Values menus.
- allow CLOS instances to display themselves, allowing graphical menu items, etc.
- there must be a rich set of facilities for running user specified code when the pointing device enters or exits a region, and when a region is "touched" by the pointing device; these "active regions" must:
  - ° be able to be non-rectangular, though they may be constrained to be rectilinear and congruent to the X and Y axis of the window system
  - ° have built in ways to work in scrolling windows
  - ° have built in ways to be highlighted via boxing, inverting, and color washing on color displays when the pointer is inside the region
- the ability to channel mouse events and keystrokes through streams
- a modular way to add items to menus in tools
- the mechanism for redisplaying windows that are being uncovered or moved should be flexible and programmable so that, for instance, windows need not have a data structure that stores the pixels associated with the window when they are not visible, a technique that can be expensive when there are many large windows or many bits per pixel
- mechanism for synchronizing keyboard and mouse so switching windows works smoothly
- have available variable width fonts

The following items are features which would be very useful, but are not required:

- optional larger or multiple displays as in the Macintosh II

- have available primitive drawing operations that are very low overhead (non-consing and fast)

- coordinate transforms

- allow access to low level color/frame buffer control for some applications

- kerning (changing the position a character is drawn in when it is next to certain other characters) of certain character combinations or at least pseudo-kerning (offsetting certain characters in a font by a small amount to improve the aesthetics of the resulting text

### 3.2. Multiple Processes

A critical part of any powerful software development environment is the ability to run multiple tasks simultaneously. In Lisp it is particularly important that one have access to multi-tasking within the single address space of the running Lisp environment so that cooperating agents can freely access shared data. These "processes" should have the following properties:

- inexpensive in terms of system resources and time to create/use (lightweight) (to assist this, it's advisable that a process not have things like an I/O window until it's needed)

- processes should be preemptable and should be scheduled under a priority and quantum based scheme

- locks and events should be available to control synchronization of cooperating agents so that, for instance, a data-producer could signal an event that would re-activate a consumer process

- the scheduler should maintain queues rather than typically calling a "are you runnable" routine to avoid high overhead when there are many processes and events should be used to move processes onto run queues

- a complete set of operations to control processes (eg. [un]arrest, kill, change priority/quantum, inspection of statistics/top-level functions/etc., and interrupt)

- the keyboard attaches to processes, not windows

- the notion of a stack or stack group should be separate from the actual process

- processes should be CLOS objects

It is conspicuous that unless process switching is a very, very low overhead operation, of the order of 2-4 function calls, the scheduler shouldn't run in it's own stack, but run on the last running processs' stack so that undue overhead isn't introduced.

It is desirable to have but potentially difficult to implement a system wherein when one process waits for an event like I/O or a page fault other processes are able to continue. We would very much like this feature, but do not require it.

### 3.3. Common Lisp

Lisp is the computing language of choice in the KSL and is likely to remain so due to it's utility in the type of programming required for the KSL's research, as well as the tremendous amount of available experience in building strong computing environments in Lisp and the strong commitment already in place.

We feel it is critical that the Lisp be Common Lisp, as described in Guy Steele's "Common Lisp the Language" today, and as specified by ANSI's X3J13 in the future.

It is essential that with the Lisp there is a strong object oriented programming system. In particular, the Common Lisp Object System {ref} (CLOS) should be fully supported, being well integrated with the Lisp support environment, and used where appropriate in system software. The CLOS specification has not yet been completed, but major parts of it are essentially complete and have been accepted by X3J13 and so, given the critical role of the object system, we feel that even an implementation of the partial specification is important, along with further implementation as the specification matures.

Similarly, we require a condition handling system, and in particular the Common Lisp Condition System {ref}, under conditions similar to CLOS. The condition system should also provide a mechanism to catch and abort "trivial" errors committed during top-level typein such as unbound variables and undefined functions and a mechanism allowing searching for functions or symbols in other packages when they are undefined (package DWIM).

The system must also support the following:

- Large FIXNUMs (at least 24 bits)
- IEEE Floating Point Numbers {what's the IEEE spec name?}

### 3.4. Compilation

Mondern Lisp systems almost always "compile" the Lisp source code into instructions more suited to the architecture of the machine in use. As execution speeds increase and the size of problems being tackled increases it's important that compilation time not introduce painful delays into the development loop, ruining the quick edit-compile-run cycle characteristic of Lisp. The compilation delay for "small" code units (approximately 10 to 50 lines of code not involving heavy macro expansion), should be negligible, while mechanisms should be available for larger "batch" compilations as part of the software management system.

It is acceptable to have mutiple ways to execute the same source code, such as an interpreter, a compiler that executes quickly but produces slower code, and a compiler that executes slowly but produces faster code. However, it is absolutely essential that all of these have indistinguishable semantics.

The following features should be present with compilation:

- Compilation of individual top-level forms (incremental compilation)
- Complete compilation of all forms, including closures and other lexical functions
- Ability to cause code to be compiled in-line via the use of declarations

- Ability to do unboxed floating point operations if appropriate declarations are present

- Documentation on built-in compiler optimizers

- The ability for the user to define new compiler optimizers

- Optimization of tail recursive calls

- Automatic compilation of "encapsulation" code such as ADVISE or TRACE (described elsewhere) so that at the time of the encapsulation the associated code is compiled

To achieve further portability, we would like to see cross compilation or multi-targeting capability, though this is not strictly required.

### 3.5. Input/Output

Moving data into and out of Lisp is something that is done quite a bit during the normal operation of the machine, so I/O performance must be on par with the rest of the system. In particular, loading compiled files (FASLOAD), reading source files into the editor, loading source files (READ), file probes (OPEN and FILE-WRITE-DATE) (typically done in the software management system) must be quick. As a guideline, we would expect read/write speeds into and out of the Lisp world of close to 40 kilobytes (kB) per second to a network file server, or 150 kB/sec to a local disk, as seen on the Explorer II. Additionally, the system should support a large number of simultaneously open files (certainly 30 or more), as well as multiple streams (input and output) to the same file. Access to remote files should be transparent to the Lisp user (ie. no special "copy to the local system" step should be needed to access data available via filing protocols including at least NFS.)

### 3.6. Utilities

There are a number of environmental utilities needed to use the system effectively, including:

- a way to save a lisp image for later reuse, as in TI's DISK-SAVE and Xerox's SYSOUT

- a mechanism for "advising"; wrapping code around entry points to affect the behavior of code as in TI Explorer Lisp's ADVISE

- routines for accessing network facilities (TCP, etc.)

- a WITH-TIMEOUT routine that would allow execution of some code to be aborted if it does not complete within the alloted time

- trace, including internals (FOO-in-BAR, LABELS, FLET, closures)

- a facility for searching the world for symbols (APROPOS)

- ways to determine what functions call another function or use a particular special variable, as well as to determine what functions are called by a (and specials used by) a particular function

- routines for laying out and drawing hierarchies and graphs where the nodes and edges can be instances that draw themselves and define their own mouse sensitivity

- a foreign language interface on machines which support non-Lisp languages

- remote procedure call (Sun RPC)

- stream interfaces to various facilities such as networks, windows, and printers

- a single stepper, probably only on interpreted code

- file properties including write-date, author, security status, locking, and properties native to the operating system; ideally the user would be able to define and use arbitrary file properties

- ability to restart Lisp in the same address space, allowing one to reinitialize windows, processes, etc. without losing edits and other work

- routines for manipulating time values

- a mechanism for reconstituting structure definitions (DEFSTRUCT)

### 3.7. Interface Toolkit

While there is a good deal of disagreement about what user interfaces should look like it is generally accepted that consistency within a system is worth working for when other considerations aren't overriding. Thus, to encourage overall consistency, we require that the higher level programming tools be built on a single interface subtrate such that they have consistent use of menus, typein, display format, etc., and that this tool be available to the user. An examples of this type of tool is TI's Universal Command Loop.

### 3.8. Help System

Important to the overall usability of the system is good "novice" support in the form of some combination of on-line help (eg. general help files), on-line tutorials (eg. the Machintosh Guided Tour), context sensitive help (including menus of commonly used commands and completion) (eg. TI's Suggestions Menus), on-line documentation (eg. doc strings in functions and variables), primer documentation, and informative error messages. The help system should be consistent, used in all the system tools, and useable in user written programs.

### 3.9. Status Information

It's important for the user, and especially, the programmer to get good information about the current operational status of the machine. The TI and Symbolics WHO-LINE and Sun's Perfmeters are examples of this sort of facility. We feel that some information about the following should be includable on the screen at all times:

- gc activity

- cpu used by lisp

- paging
- consing
- system load
- current package
- status of process owning the keyboard
- what process owns the keyboard
- lisp file activity

### 3.10. Printing

While the capabilities of the system being specified will encourage a paperless office, hardcopy printing is still an important part of our activities for debugging, passing along information, and keeping records. The printing system should:

- use generic operations as in Xerox Lisp (see window system discussion)
- allow users to add new printer types/drivers
- support at least PostScript initially
- allow printing of unformatted files, formatted files, and window images

### 3.11. Pretty Printing

The ability of the system to format output, especially in a window based environment, is important to the user's ability to understand the data being displayed. Thus, a "pretty printing" facility must be included with the following features:

- user customizable
- has a protocol to interact with instances so that they can make formatting decisions
- interprets arg lists for macros and formats accordingly so that, for instance, &BODY arguments get formatted as code
- works with intermediate data structures so that entire expressions need not be printed for efficiency with scrolling windows, especially in the inspector
- can format user-defined mouse sensitive data

## 4. Lower Level Issues

### 4.1. Address Space

As the size of problems being addressed increases so does the need for address space in Lisp systems. It is difficult to quantify address space requirements as they are affected by other facets of the system including effectiveness of the memory management system and the space required by data strucutures, but we can say that we need the potential for a very large address space, such that the address space is typically limited by how much disk it's feasible to have rather than the number of bits used in addresses. Good examples of today's systems are the Symbolics 3600 {number} or the TI Explorer with Extended Address Spaces {number}. Note that systems that migrate unused objects out of the primary address space should allow a primary address space of at least 100 megabytes. We expect this requirement to expand in the future.

Conditions should be signalled when address space gets to a user-definable minimum, with default handlers that will notify the user of the low address space condition.

A parallel to address space is stack size. Recursive or other deeply nested programs must work without modification and so we require that the execution stack be expandable at run time as with the TI Explorer and Symbolics systems, or very, very large. Minimally the stack should be large enough to run 5,000 function call levels with an average of 4 arguments and 4 locals per level in the most stack-hungry execution mode (usually interpreted).

### 4.2. Memory Management

A frequent thorn in the side of Lisp programmers is the reclamation of allocated but no longer used memory, or garbage collection (GC). Therefore, we require that:

- in general, GC take no more than 10% total overhead, with less being very desirable

- no programmer/user intervention be required in normal operation

- the amount of time that the machine is made unavailable to the user is limited to a few seconds at a time either by time limiting the amount of work done at once or by using a concurrent system, with either solution implying a dynamic algorithm

- the working set not be unduly expanded by GC operation to avoid thrashing

- there be controls available to the programmer to tune the GC to a particular program, or to inhibit it at times for real-time program segments or for timing

- there be finalization code associated with some objects like CLOS instances for cases like the need to release resources that aren't resident in the Lisp address space

In addition to the automatic memory management software there should be tools that allow a programmer explicit control over storage allocation with notions similar to "area"s for new allocation which can be declared exempt from GC or to be deallocated in bulk. Also allocation aids such as RESOURCE structures should be provided.

### 4.3. Dedicated Versus Shared Systems

Timesharing, as opposed to having a processor dedicated to a single user, is acceptable in principle, though it is important that it be well done in the sense that users not step on each others' toes by doing simple things like running programs. In particular the scheduler and paging algorithms should be such that if the system is claimed to support N users, all N users should simultaneously see the kind of minimum responsiveness and performance we require.

### 4.4. Hardware Capabilities

The display should be able show approximately 70 lines of monochrome text with 130 columns each and still be read comfortably, such as the approximately 1024 x 768, 72 dot/inch black and white displays found on TI Explorers and Symbolics 3600 class machines, with a strong desire for being able to display two 80 character wide windows side by side, as Sun workstations with hi-resolution displays and Xerox 1186's with 19" displays. The display must be stable and crisp, which means it should probably be non-interlaced. It must also be possible to get a video output for demonstrating software to large audiences.

The data input mechanism should support a rate at least equal to that found in accomplished touch typists (approximately 70 words per minute), such as a keyboard with 2 key or more rollover, as well as a fast pointing device equipped with at least two, and preferably three kinds of "touches", such as a 3 button mouse, a way of programming idioms into short cut sequences, such as programmable function keys on a keyboard, and a way of sending non-text commands which would mean at least two (Control and Meta) modifier keys on a keyboard.

### 4.5. Overall System Integration in the KSL

Any new computing systems in the KSL must be able to fit in to the existing environment and interact with pre-existing systems to facilitate sharing data, moving users from system to system, and system administration. Incoming systems should have the following properties to integrate well into the KSL environment:

- good networking including filing (Sun IP/UDP/NFS and IP/TCP/FTP minimally), virtual terminal service (IP/TCP/TELNET), remote procedure call (Sun IP/UDP/RPC), and name service (IP/UDP/DOMAIN)

- provisions for file backup, possibly via NFS

- a large limit, if any, on file names, with 40 characters per field minimum

- if the system is a workstation, the user must be able to reboot from the console and be able to run first-order diagnostics to determine in most cases what major component is responsible for any failure

- the element of the system that sits in offices should have minimal power requirements, thus requiring no additional air conditioning capacity, and should not generate distracting noise; if the system is a workstation unit, it is probably necessary to remote the processor from the display to achieve this requirement and we would need at least 500 feet of potential separation to reach from our office spaces to our machine room spaces; for example, we consider the TI Explorer too noisy and hot to have the system unit in most offices, and consider most Apple Macintosh II's to be just under the acceptable noise level

## 5. Acknowledgement

The summary presented here is the result of extensive discussions among the following group: Bruce Delagi, Christopher Lane, James Rice, Thomas Rindfleisch, and Eric Schoen.

# Appendix D

# AIM Management Committee Membership

Following are the current membership lists of the various SUMEX-AIM management committees:

*AIM Executive Committee:*

SHORTLIFFE, Edward H., M.D., Ph.D. (Chairman)
Principal Investigator - SUMEX
Medical School Office Building, Rm. X271
Stanford University Medical Center
Stanford, California 94305
(415) 723-6970

FEIGENBAUM, Edward A., Ph.D.
Co-Principal Investigator - SUMEX
Heuristic Programming Project
Department of Computer Science
701 Welch Road, Building C
Stanford University
Stanford, California 94305
(415) 723-4879

KULIKOWSKI, Casimir, Ph.D.
Department of Computer Science
Rutgers University
New Brunswick, New Jersey 08903
(201) 932-2006

LEDERBERG, Joshua, Ph.D.
President
The Rockefeller University
1230 York Avenue
New York, New York 10021
(212) 570-8080, 570-8000

LINDBERG, Donald A.B., M.D. (Past Adv Group Chrmn)
Director, National Library of Medicine
8600 Rockville Pike
Bethesda, Maryland 20814
(301)496-6221

MYERS, Jack D., M.D.
School of Medicine
Scaife Hall, 1291
University of Pittsburgh
Pittsburgh, Pennsylvania 15261
(412) 648-9933

*AIM Advisory Group:*

MYERS, Jack D., M.D. (Chairman)
School of Medicine
Scaife Hall, 1291
University of Pittsburgh
Pittsburgh, Pennsylvania 15261
(412) 648-9933

AMAREL, Saul, Ph.D.
Department of Computer Science
Rutgers University
New Brunswick, New Jersey 08903
(201) 932-3546

COULTER, Charles L., Ph.D. (Exec. Secretary)
Bldg 31, Room 5B41
Biomedical Research Technology Program
National Institutes of Health
9000 Rockville Pike
Bethesda, Maryland 20892
(301) 496-5411

FEIGENBAUM, Edward A., Ph.D. (Ex-officio)
Co-Principal Investigator - SUMEX
Heuristic Programming Project
Department of Computer Science
701 Welch Road, Building C
Stanford University
Palo Alto, California 94305
(415) 723-4879

KULIKOWSKI, Casimir, Ph.D.
Department of Computer Science
Hill Center Busch Campus
Rutgers University
New Brunswick, New Jersey 08903
(201) 932-2006

LEDERBERG, Joshua, Ph.D.
President
The Rockefeller University
1230 York Avenue
New York, New York 10021
(212) 570-8080, 570-8000

LINDBERG, Donald A.B., M.D.
Director, National Library of Medicine
Building 38, Rm. 2E-17B
8600 Rockville Pike
Bethesda, Maryland 20814
(301) 496-6221

MINSKY, Marvin, Ph.D.
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
545 Technology Square
Cambridge, Massachusetts 02139
(617) 253-5864

MOHLER, William C., M.D.
Associate Director
Division of Computer Research and Technology
National Institutes of Health
Building 12A, Room 3033
9000 Rockville Pike
Bethesda, Maryland 20892
(301) 496-1168

PAUKER, Stephen G., M.D.
Department of Medicine - Cardiology
Tufts New England Medical Center Hospital
171 Harrison Avenue
Boston, Massachusetts 02111
(617) 956-5910

SHORTLIFFE, Edward H., M.D., Ph.D. (Ex-officio)
Principal Investigator - SUMEX
Medical School Office Building, Rm. X271
Stanford University Medical Center
Stanford, California 94305
(415) 723-6979

SIMON, Herbert A., Ph.D.
Department of Psychology
Baker Hall, 339
Carnegie-Mellon University
Schenley Park
Pittsburgh, Pennsylvania 15213
(412) 578-2787, 578-2000

*Stanford Community Advisory Committee:*

SHORTLIFFE, Edward H., M.D., Ph.D. (Chairman)
Principal Investigator - SUMEX
Medical School Office Building, Rm. X271
Stanford University Medical Center
Stanford, California 94305
(415) 723-6979

FEIGENBAUM, Edward A., Ph.D.
Heuristic Programming Project
Department of Computer Science
Margaret Jacks Hall
Stanford University
Stanford, California 94305
(415) 723-4879

LEVINTHAL, Elliott C., Ph.D.
Departments of Mechanical and Electrical Engineering
Building 530
Stanford University
Stanford, California 94305
(415) 723-9037

# Appendix E

# Scientific Subproject Abstracts

The following are brief abstracts of our collaborative research projects.

National AIM Project: ATTENDING Project:
A Critiquing Approach to
Expert Computer Advice

Principal Investigator: Perry L. Miller, M.D., Ph.D.
Department of Anesthesiology
Yale University School of Medicine
New Haven, CT 06510
(203) 785-2802

Our project is exploring the "critiquing" approach to bringing computer-based advice to the practicing physician.

Critiquing is a different approach to the design of artificial intelligence based expert systems. Most medical expert systems attempt to simulate a physician's decision-making process. As a result, they have the clinical effect of trying to tell a physician what to do: how to practice medicine. In contrast, a critiquing system first asks the physician how he contemplates approaching his patient's care, and then critiques that plan. In the critique, the system discusses any risks or benefits of the proposed approach, and of any other approaches which might be preferred. It is anticipated that the critiquing approach may be particularly well suited for domains, like medicine, where decisions involve a great deal of *subjective* judgment.

To date, severa; prototype critiquing systems have been developed in different medical domains:

1. ATTENDING, the first system to implement the critiquing approach, critiques anesthetic management.
2. HT-ATTENDING critiques the pharmacologic management of essential hypertension.
3. VQ-ATTENDING critiques aspects of ventilator management.
4. PHEO-ATTENDING critiques the laboratory and radiologic workup of a patient for a suspected pheochromocytoma.

In addition, a domain-independent system, ESSENTIAL-ATTENDING, has been developed to facilitate the implementation of critiquing systems in other domains.

PUBLICATIONS

1. Miller, P.L. (Ed.): Selected Topics in Medical Artificial Intelligence. New York: Springer-Verlag (in press).
2. Rennels, G.D., Miller, P.L.: Artificial intelligence research in anesthesia and intensive care. Journal of Clinical Monitoring (in press).
3. Miller, P.L., Rennels, G.D.: Prose generation from expert systems: An applied computational linguistics approach. AI Magazine (in press).
4. Rennels, G.D., Shortliffe, E.H., Stockdale, F.E., Miller, P.L.: A computational model of reasoning from the clinical literature. Computer Methods and Programs in Biomedicine 24:139-149, 1987.
5. Rennels, G.D., Shortliffe, E.H., Stockdale, F.E., Miller, P.L.: A structured representation of the clinical literature and its use in a medical management advice system. Bulletin du Cancer 74:215-220, 1987.

6. Miller, P.L., Barwick, K.W., Morrow, J.S., Powsner, S.M., Riely, C.A.: Semantic relationships and medical bibliographic retrieval: A preliminary assessment. Computers and Biomedical Research 21:64-77, 1988.

7. Miller, P.L., Morrow, J.S., Powsner, S.M., Riely, C.A.: Semantically assisted medical bibliographic retrieval: An experimental computer system. Bulletin of the Medical Library Association 76:131-136, 1988.

8. Miller, P.L.: Exploring the critiquing approach: Clinical practice-based feedback by computer. Biomedical Measurement, Informatics and Control (in press).

9. Rennels, G.D., Shortliffe, E.H., Stockdale, F.E., Miller, P.L.: A computational model of reasoning from the clinical literature. The AI Magazine (accepted pending revision).

10. Miller, P.L., Fisher, P.R.: Causal models in medical artificial intelligence. Proceedings of the Eleventh Symposium on Computer Applications in Medical Care, Washington, D.C., November 1987, pp. 17-22.

11. Powsner, S.M., Barwick, K.W., Morrow, J.S., Riely, C.A., Miller, P.L.: Coding semantic relationships for medical bibliographic retrieval: A preliminary study. Proceedings of the Eleventh Symposium on Computer Applications in Medical Care, Washington, D.C., November 1987, pp. 108-112.

Stanford Project: BBICU -- BLACKBOARD APPLICATIONS IN THE INTENSIVE CARE UNIT

Principal Investigator:

Adam Seiver, M.D.
Department of Surgery
Palo Alto Veterans Medical Center
Palo Alto, Calif

Lawrer⁻e Fagan, M.D., Ph.D.
Department of Medicine
Stanford University

Barbara Hayes-Roth, Ph.D.
Department of Computer Science
Stanford University

We are designing a data-interpretation and therapy-planning system for the intensive care unit (ICU). Fundamental research issues in temporal reasoning are associated with the ICU application area including assimilation of incoming data, representation of time-oriented intervals, and description of ongoing physiological processes [Fagan 84]. In addition, in ICUs of the 1990s, many more physiological measurements will need to be collected at frequent intervals, and increased access to the current medical record in coded format will be possible. Processing of incoming data will have to be opportunistic, selecting from a number of models that have different computational requirements and accuracy. We will use a blackboard architecture, known as BB1, that has evolved from earlier work on protein-structure elucidation and construction layout [Hayes-Roth 85]. BB1 is particularly well suited for the ICU project because it maintains separate blackboards for domain and control knowledge.

Although we have selected the blackboard structure as the organizing principle, many knowledge representation issues remain. First, we must represent the structure and function (anatomy and physiology) of the respiratory system. By characterizing the pathophysiology in terms of generic faults, we will create a more flexible means to diagnose problems in unusual situations -- in contrast to the phenomenological rules used in earlier systems.

Second, we must coalesce quantitative and qualitative models. The physiology of the respiratory and cardiac systems have been modeled in detail, but it is impractical to base the entire reasoning process on complex mathematical equations. Instead, we are developing methods to transform quantitative models into simpler formulations. We must make explicit the simplifying assumptions and associate them with their corresponding models, so that we can select an applicable model for situations of varying complexity.

Using the approach of our project for planning treatments for cancer patients [Langlotz 87], we will use strategic knowledge to create patient-specific specializations of standard treatment plans. We will use decision analytic methods to evaluate and explain the various treatment options available at any point in time. The long-term goal of this project is to embed the decision-making components within the data management tasks of the ICU.

## REFERENCES

1. Fagan, L., Kunz, J., Feigenbaum, E, and Osborn, J. Adapting a rule-based system for a monitoring task, in *Rule Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project*, B. Buchanan and E. Shortliffe (eds.). Reading, MA: Addison-Wesley Publishing Co., 1984.

2. Hayes-Roth, B. A Blackboard architecture for control. (Artificial Intelligence) 26:251-321, 1985.

3. Langlotz, C., Fagan, L., Tu, S., Sikic, B., and Shortliffe, E. A therapy planning architecture that combines decision theory and artificial intelligence techniques. *Computers and Biomedical Research* 20:279-303, 1987.

National AIM Project: DECISION SUPPORT FOR
TIME-VARYING CLINICAL PROBLEMS

Principal Investigator: Lawrence Widman, M.D., Ph.D.
Division of Cardiology
University of Texas Health Science Center
San Antonio, Texas 78284

Time-varying systems, which include many areas of medicine, science, economics, and business, can be described mathematically by differential equations. They are distinct from the pattern-matching and logic-based domains dealt with so successfully by existing expert system methods, because they can include feedback relationships. It is generally felt that they are best approached by enhancement of existing methods for deep model-based reasoning.

The goal of this project is to develop AI methods for capturing and using knowledge about time-varying systems. The strategy is to address general problems in model-based knowledge representation and reasoning. The intermediate objective is to develop methods which are powerful enough to work in selected realistic situations yet are general enough to be transportable to other, unrelated knowledge domains.

The tactical approach is to work on well-defined yet complex and interesting problems in the medical domain. We have, therefore, selected the human cardiovascular system as our prototype of a time-varying system, and are developing methods for representing and reasoning about its mechanical and electrical activities in the normal and diseased states.

REFERENCES

1. Widman, L.E. Knowledge-Based Fault Identification and "What If" Simulation in Symbolic Dynamic Systems Models. Proceedings of the 1988 Society for Computer Simulation Multiconference on AI and Simulation, pp. 89-94, 1988.

2. Widman, L.E., Lee, Y.-B., and Y.-H. Pao. Diagnosis of Causal Medical Models by Semi-Quantitative Reasoning. In: Miller, P.L. (ed.). Topics in Medical Artificial Intelligence, Springer-Verlag (in press)

3. Widman, L.E. Semi-Quantitative "Close Enough" Systems Dynamics Models: An Alternative to Qualitative Simulation. In: Widman, L.E., Loparo, K.A., and N.R. Nielsen (eds.). Artificial Intelligence, Simulation, and Modeling. John Wiley & Sons, New York (in press).

4. Widman, L.E., Loparo, K.A., and N.R. Nielsen (eds.). Artificial Intelligence, Simulation, and Modeling. John Wiley & Sons, New York (in press).

5. Widman, L.E. and G.L. Freeman. A-to-D Conversion from Paper Records with a Desktop Scanner and a Microcomputer. (submitted to the American Journal of Physiology).

Stanford Project: GUIDON/NEOMYCIN --
KNOWLEDGE ENGINEERING
FOR TEACHING MEDICAL DIAGNOSIS

Principal Investigators:

William J. Clancey, Ph.D.
701 Welch Road
Department of Computer Science
Stanford University
Palo Alto, California 94304
(415) 723-1997 (CLANCEY@SUMEX-AIM)

Bruce G. Buchanan, Ph.D.
Computer Science Department
Stanford University

The GUIDON/NEOMYCIN Project is a research program devoted to the development of a knowledge-based tutoring system for application to medicine. The key issue for the GUIDON/NEOMYCIN project is to develop a program that can provide advice similar in quality to that given by human experts, modeling how they structure their knowledge as well as their problem-solving procedures. The consultation program using this knowledge is called NEOMYCIN. NEOMYCIN's knowledge base, designed for use in a teaching application, is the subject material used by a family of instructional programs referred to collectively as GUIDON2. The problem-solving procedures are developed by running test cases through NEOMYCIN and comparing them to expert behavior. Also, we use NEOMYCIN as a test bed for the explanation capabilities incorporated in our instructional programs.

The goal of current work is to construct a knowledge-based tutoring system that teaches diagnostic strategies explicitly. By strategy, we mean plans for establishing a set of possible diagnoses, focusing on and confirming individual diagnoses, gathering data, and processing new data. The tutorial program has capabilities to recognize these plans, as well as to articulate strategies in explanations about how to do diagnosis. The strategies represented in the program, modeling techniques, and explanation techniques are wholly separate from the knowledge base, so that they can be used with many medical (and non-medical) domains. That is, the target program will be able to be tested with other knowledge bases, using system-building tools that we provide.

SOFTWARE AVAILABLE ON SUMEX

GUIDON--A system developed for intelligent computer-aided instruction. Although it was developed in the context of MYCIN's infectious disease knowledge base, the tutorial rules will operate upon any EMYCIN knowledge base.

NEOMYCIN--A consultation system derived from MYCIN, with the knowledge base greatly extended and reconfigured for use in teaching. In contrast with MYCIN, diagnostic procedures, common sense facts, and disease hierarchies are factored out of the basic finding/disease associations. The diagnostic procedures are abstract (not specific to any problem domain) and model human reasoning, unlike the exhaustive, top-down approach implicit in MYCIN's medical rules. This knowledge base is used in the GUIDON2 family of instructional programs, being developed on D-machines.

REFERENCES

1. Clancey, W.J.: **Knowledge-Based Tutoring: The GUIDON Program**, Cambridge: The MIT Press, 1987.

2. Clancey, W.J.: *Methodology for building an intelligent tutoring system.* In Kintsch, Polson, and Miller, (Eds.), METHODS AND TACTICS IN COGNITIVE SCIENCE. L. Erlbaum Assoc., Hillsdale, NJ. 1984. (Also STAN-CS-81-894, HPP 81-18)

3. Clancey, W.J.: *Acquiring, representing, and evaluating a competence model of diagnosis.* In Chi, Glaser, and Farr (Eds.), THE NATURE OF EXPERTISE. 1988.

National AIM Project: INTERNIST-I/QMR (Quick Medical Reference)

Principal Investigators:

INTERNIST-QMR PROJECT:
Randolph A. Miller, M.D. (RMILLER@SUMEX-AIM)
Jack D. Myers, M.D. (MYERS@SUMEX-AIM)
Fred E. Masarie, Jr., M.D. (MASARIE@SUMEX-AIM)
University of Pittsburgh
Pittsburgh, Pennsylvania 15261
(412) 648-3190

The major goal of the INTERNIST-I/QMR Project is to produce a reliable and adequately complete diagnostic consultative program in the field of internal medicine. Although this program is intended primarily to aid skilled internists in complicated medical problems, the program may have spin-offs as a diagnostic and triage aid to physicians' assistants, rural health clinics, military medicine and space travel. In the design of INTERNIST-I and QMR, we have attempted to model the creative, problem-formulation aspect of the clinical reasoning process, and to design program components that can help physicians at key points in this process. The program can compose differential diagnoses, dynamically, on the basis of clinical evidence. During the course of an INTERNIST-I consultation, it is not uncommon for a number of such conjectured problem foci to be proposed and investigated. With the QMR program, the physician is given more control over the direction of the problem-solving activities than was possible with INTERNIST-I. QMR is broader in scope than INTERNIST-I in that it provides quick and efficient access to the INTERNIST-I/QMR knowledge base to provide low and intermediate levei informational support for physicians' decision-making, in addition to providing consultative advice.

SOFTWARE AVAILABLE ON SUMEX

Versions of the INTERNIST-I KB are available for experimental use, with permission from the developers. In the past year, copies have been shared with Drs. Gregory Cooper and Homer Chin of Dr. Shortliffe's group at Stanford. The INTERNIST-I and QMR knowledge bases are Copyright 1986-88 by the University of Pittsburgh. Quick Medical Reference and QMR are registered trademarks of the University of Pittsburgh. The project continues to be oriented primarily towards research and development; hence, a stable production version of the system is not yet available for general use. QMR has been shared on a restricted basis with a limited number of academic colleagues, who have agreed to give the QMR development team feedback on the program's strengths and weaknesses.

REFERENCES

1. Myers JD. The Background of INTERNIST-I and QMR. In: Proceedings of the History of Medical Informatics Conference. National Library of Medicine. pp. 195-197, November 1987.

2. Masarie, F.E., Miller, R.A. Medical Subject Headings and Medical Terminology: An analysis of terminology used in hospital charts. Bulletin of the Medical Library Association, 1987; 75:89-94.

3. Masarie FE, Miller RA. INTERNIST-I to Quick Medical Reference (QMR): Transition from a mainframe to a microcomputer. Proceedings Ninth Annual IEEE/Engineering in Biology and Biology Society. IEEE Press. pp. 1521- 1522, November 1987.

4. Parker, RC, Miller RA. Using causal knowledge to create simulated

patient cases: The CPCS project as an extension of INTERNIST-I. Proceedings of the Eleventh Annual Symposium on Computer Applications in Medical Care. pp. 473-480, November 1987.

5. Bankowitz RA, Blumenfeld BH, Miller RA, et al. User variability in abstracting and entering printed case histories with Quick Medical Reference (QMR). Proceedings of the Eleventh Annual Symposium on Computer Applications in Medical Care. pp. 68-73, November 1987.

6. Masarie FE, Miller RA. Quick Medical Reference (QMR): An information management tool for clinical diagnosis. IN: Critical Reviews in Medical Informatics. Boca Raton, Florida, CRC Press, 1988.

7. Parker RC, Miller RA. Using causal knowledge to created simulated patient cases: The CPCS project as an extension of INTERNIST-I. IN: Miller PL (ed) Topics in Medical Artificial Intelligence. Computers and Medicine Series, Springer-Verlag, New York, 1988.

8. Challinor SM, McNeil MA, Bankowitz RA, et al. Evaluation of a Computer-Assisted General Medicine Diagnostic Consultation Service. Abstract presented at the 11th Annual SGIM Meeting, Washington, D.C., April 27-29, 1988.

9. Miller RA. From Automated Medical Records to Expert System Knowledge Bases: Common Problems in Representing and Processing Patient Data. Topics in Health Record Management. 75(3):23-36, 1987

10. Miller RA. Computer-based Diagnostic Decisionmaking. Medical Care. 25(12):S148-S152, 1987.

National AIM Project: MENTOR -- MEDICAL EVALUATION OF THERAPEUTIC ORDERS

Principal Investigators:

Stuart Speedie, Ph.D. (SPEEDIE@SUMEX-AIM)
School of Pharmacy
University of Maryland
20 N. Pine Street
Baltimore, Maryland 21201
(301) 528-7650

Terrence F. Blaschke, M.D. (BLASCHKE@SUMEX-AIM)
Department of Medicine
Division of Clinical Pharmacology
Stanford University Medical Center
Stanford, California 94305

The goal of the MENTOR project is to implement and begin evaluation of a computer-based methodology for reducing therapeutic misadventures. The project uses an on-line expert system to continuously monitor the drug therapy of individual patients and generate specific warnings of potential and/or actual unintended effects of therapy. The appropriate patient information is automatically acquired through interfaces to a hospital information system. This data is monitored by a system that is capable of employing complex chains of reasoning to evaluate therapeutic decisions and arrive at valid conclusions in the context of all information available on the patient. The results reached by the system are fed back to the responsible physicians to assist future decision making.

Specific objectives of this project include:

1. Implement a prototype computer-based expert system to continuously monitor in-patient drug therapy that uses a modular medical knowledge base and a separate inference engine to apply the knowledge to specific situations.

2. Select a small number of important and frequently occurring drug therapy problems that can lead to therapeutic misadventures and construct a comprehensive knowledge base necessary to detect these situations.

3. Evaluate the prototype MENTOR system with respect to its impact on the on the physicians' therapeutic decision making as well as its effects on the patient in terms of specific mortality and morbidity measures.

The work in this project builds on the extensive previous work in drug monitoring done by these investigators in the Division of Clinical Pharmacology at Stanford and the University of Maryland School of Pharmacy.

Stanford Project: MOLGEN -- AN EXPERIMENT PLANNING SYSTEM FOR MOLECULAR GENETICS

Principal Investigators:

Edward A. Feigenbaum, Ph.D.
Department of Computer Science
Stanford University

Charles Yanofsky, Ph.D. (YANOFSKY@SUMEX-AIM)
Department of Biology
Stanford University
Stanford, California 94305
(415) 725-3815

The MOLGEN project has focused on research into the applications of symbolic computation and inference to the field of molecular biology. This has taken the specific form of systems which provide assistance to the experimental scientist in various tasks, the most important of which have been the design of complex experiment plans and the analysis of nucleic acid sequences. Our current research concentrates on scientific discovery within the subdomain of regulatory genetics. We desire to explore the methodologies scientists use to modify, extend, and test theories of genetic regulation, and then emulate that process within a computational system.

Theory or model formation is a fundamental part of scientific research. Scientists both use and form such models dynamically. They are used to predict results (and therefore to suggest experiments tc test the model) and also to explain experimental results. Models are extended and revised both as a result of logical conclusions from existing premises and as a result of new experimental evidence.

Theory formation is a difficult cognitive task, and one in which there is substantial scope for intelligent computational assistance. Our research is toward building a system which can form theories to explain experimental evidence, can interact with a scientist to help to suggest experiments to discriminate among competing hypotheses, and can then revise and extend the growing model based upon the results of the experiments.

The MOLGEN project has continuing computer science goals of exploring issues of knowledge representation, problem-solving, discovery, and planning within a real and complex domain. The project operates in a framework of collaboration between the Heuristic Programming Project (HPP) in the Computer Science Department and various domain experts in the departments of Biochemistry, Medicine, and Biology. It draws from the experience of several other projects in the HPP which deal with applications of artificial intelligence to medicine, organic chemistry, and engineering.

## SOFTWARE AVAILABLE ON SUMEX

SPEX system for experiment design.

UNITS system for knowledge representation and acquisition.

SEQ system for nucleotide sequence analysis.

## REFERENCES

1. Friedland, P.E.: *Knowledge-based experiment design in molecular genetics,* (Ph.D. thesis). Stanford Computer Science Report, STAN-CS-79-771.

2. Friedland, P.E. and Iwasaki, Y.: *The concept and implementation of skeletal plans*, Journal of Automated Reasoning, 1(2):161-208, 1985.

3. Friedland, P.E. and Kedes, L.: *Discovering the secrets of DNA*, Communications of the ACM, 28(11):1164-1186, November, 1985.

4. Stefik, M.J.: *An examination of a frame-structured representation system*, Proc. Sixth IJCAI, Tokyo, August, 1979, pp. 845-852.

5. Stefik, M.J.: *Planning with constraints*, (Ph.D. thesis). Stanford Computer Science Report, STAN-CS-80-784, March, 1980.

6. Karp, P., and D. Wilkins: *An Analysis of the Deep/Shallow Distinction for Expert Systems*. Stanford University Knowledge Systems Laboratory Report KSL-86-32, 1986.

7. Karp, P., and P. Friedland: *Coordinating the Use of Qualitative and Quantitative Knowledge in Declarative Device Modeling*. Stanford University Knowledge Systems Laboratory Report KSL-87-09, 1987.

8. Round, A.: *QSOPS: A Workbench Environment for the Qualitative Simulation of Physical Processes*. Stanford University Knowledge Systems Laboratory Report KSL-87-37, 1987.

9. Karp, P.: *A Process-Oriented Model of Bacterial Gene Regulation*, Stanford University Knowledge Systems Laboratory Technical Report KSL-88-18.

Stanford Project: ONCOCIN -- KNOWLEDGE ENGINEERING FOR ONCOLOGY CHEMOTHERAPY CONSULTATION

Principal Investigator: Edward H. Shortliffe, M.D., Ph.D.
Departments of Medicine and Computer Science
Stanford University Medical Center
Medical School Office Building
Stanford, California 94305
(415) 723-6979 (SHORTLIFFE@SUMEX-AIM)

Project Director: Dr. Lawrence M. Fagan (FAGAN@SUMEX-AIM)

The ONCOCIN Project is overseen by a collaborative group of physicians and computer scientists who are developing an intelligent system that uses the techniques of knowledge engineering to advise oncologists in the management of patients receiving cancer chemotherapy. The general research foci of the group members include knowledge acquisition, inexact reasoning, explanation, and the representation of time and of expert thinking patterns. Much of the work developed from research in the 1970's on the MYCIN and EMYCIN programs, early efforts that helped define the group's research directions for the coming decade. MYCIN and EMYCIN are still available on SUMEX for demonstration purposes.

The prototype ONCOCIN system is in limited experimental use by oncologists in the Stanford Oncology Clinic. Thus, much of the emphasis of this research has been on human engineering so that the physicians will accept the program as a useful adjunct to their patient care activities. ONCOCIN has generally been well-accepted since its introduction, and we are now testing a version of the program which runs on professional workstations (rather than the central SUMEX computer) so that it can be implemented and evaluated at sites away from the University.

Significant extensions to the basic ONCOCIN project have included the OPAL oncology knowledge acquisition system, the PROTEGE system for description of knowledge acquisition tools for clinical trial application areas, and the ONYX system for strategic therapy planning.

SOFTWARE AVAILABLE ON SUMEX

MYCIN-- A consultation system designed to assist physicians with the selection of antimicrobial therapy for severe infections. It has achieved expert level performance in formal evaluations of its ability to select therapy for bacteremia and meningitis. Although MYCIN is no longer the subject of an active research program, the system continues to be available on SUMEX for demonstration purposes and as a testing environment for other research projects.

EMYCIN-- The "essential MYCIN" system is a generalization of the MYCIN knowledge representation and control structure. It is designed to facilitate the development of new expert consultation systems for both clinical and non-medical domains.

ONCOCIN-- This system is in clinical use but requires Lisp machines to be run. Much of the knowledge in the domain of cancer chemotherapy is already well-specified in protocol documents, but expert judgments also need to be understood and modeled.

REFERENCES

1. Shortliffe, E.H., Scott, A.C., Bischoff, M.B., Campbell, A.B., van Melle, W. and Jacobs, C.D.: *ONCOCIN: An expert system for oncology protocol management.* Proc. Seventh IJCAI, pp. 876-881, Vancouver, B.C., August, 1981.

2. Duda, R.O. and Shortliffe, E.H.: *Expert systems research.* Science 220:261-268, 1983.

3. Langlotz, C.P. and Shortliffe, E.H.: *Adapting a consultation system to critique user plans.* Int. J. Man-Machine Studies 19:479-496, 1983.

4. Bischoff, M.B., Shortliffe, E.H., Scott, A.C., Carlson, R.W. and Jacobs, C.D.: *Integration of a computer-based consultant into the clinical setting.* Proceedings 7th Annual Symposium on Computer Applications in Medical Care, pp. 149-152, Baltimore, Maryland, October 1983.

5. Hickam, D.H., Shortliffe, E.H., Bischoff, M.B., Scott, A.C., Jacobs, C.D.: A study of the treatment advice of a computer-based cancer chemotherapy protocol advisor (Memo KSL-85-21). *Annals of Internal Medicine* 103(6 pt 1):928-936 (1985).

6. Kent, D.L., Shortliffe, E.H., Carlson, R.W., Bischoff, M.B., Jacobs, C.D.: Improvements in data collection through physician use of a computer-based chemotherapy treatment consultant (Memo KSL-85-22). *Journal of Clinical Oncology* 3:1409-1417, 1985.

7. Tsuji, S. and Shortliffe, E.H.: Graphical access to a medical expert system: I. Design of a knowledge engineer's interface (Memo KSL-85-11). Meth. Inf. Med., 25:62-70, 1986.

8. Lane, C.D., Differding, J.C., Shortliffe, E.H.: Graphical access to a medical expert system: II. Design of an interface for physicians (Memo KSL-85-15). Meth. Inf. Med., 25:143-150, 1986.

9. Shortliffe, E.H. Medical expert systems: Knowledge tools for physicians. Memo KSL-86-52. Special issue on Medical Informatics, West. J. Med. 145:830-839, 1986.

10. Walton, J.D., Musen, M.A., Combs, D., Lane, C.D., Shortliffe, E.H., and Fagan, L.M. Graphical access to medical expert systems: III. Design of a knowledge-acquisition environment. Memo KSL-85-30. Methods of Information in Medicine, 26:78-88, 1987.

11. Langlotz, C.P., Fagan, L.M., Tu, S.W., Sikic, B.I., and Shortliffe, E.H. A therapy planning architecture that combines decision theory and artificial intelligence techniques. KSL-85-55. Computers in Biomedical Research, 20:279-303, 1987.

12. Musen, M.A., Fagan, L.M., Combs, D.M., and Shortliffe, E.H. Use of a domain model to drive an interactive knowledge-editing tool (Memo KSL-86-24). International Journal of Man-Machine Studies 26(1):105-121, 1987.

National AIM Project: Computer-Aided Diagnosis of Lymph Node Pathology (PATHFINDER)

Principal Investigator: Bharat Nathwani, M.D.
Department of Pathology
HMR 204
2025 Zonal Avenue
University of Southern California
School of Medicine
Los Angeles, California 90033
(213) 226-7064 (NATHWANI@SUMEX-AIM)

The Pathfinder Project is centered on the construction of an expert system for assisting pathologists with the diagnosis of tissue pathology. Pathfinder research is focused on the domain of lymph node pathology. The project is based at the University of Southern California in collaboration with the Stanford University Medical Computer Science Group. Ongoing AIM research has been addressing fundamental problems of knowledge representation, reasoning strategies, user modeling, explanation, and user acceptance. The central focus of research has been on developing tractable knowledge acquisition and reasoning methods for probabilistic reasoning. In addition to the ongoing theoretical research, the team have implemented an expert system program and a tool for knowledge acquisition. The Pathfinder expert system provides diagnostic advice on eighty common benign and malignant diseases of the lymph nodes based on 150 histologic features. Currently a more sophisticated knowledge base that represents probabilistic dependencies is being developed. A probabilistic knowledge acquisition tool has been implemented on the Macintosh that enables a user to assess probabilistic relationships by graphically noting similarities and differences between diseases hypotheses.

SOFTWARE AVAILABLE ON SUMEX

Pathfinder -- A pilot version of the Pathfinder program is available for experimentation on the DEC 2060 computer. This version is an early version of the program that has not been completely tested.

REFERENCES

1. Horvitz, E.J., Heckerman, D.E., Nathwani, B.N. and Fagan, L.M.: *Diagnostic Strategies in the Hypothesis-directed Pathfinder System, Node Pathology*. HPP Memo 84-13. Proceedings of the First Conference on Artificial Intelligence Applications, Denver, Colorado, Dec., 1984.

2. Heckerman, D. E., and Horvitz, E. J., "The Myth of Modularity in Rule-based Systems," in Uncertainty in Artificial Intelligence, Vol. 2, J. Lemmer, L. Kanal, ed., North Holland, New York, 1987.

3. Horvitz, E.J., Heckerman, D.E., Nathwani, B.N. and Fagan, L.M.: *The Use of a Heuristic Problem-solving Hierarchy to Facilitate the Explanation of Hypothesis-directed Reasoning*. KSL Memo 86-2. Proceedings of MedInfo, Washington D.C., October, 1986.

4. Horvitz, E. J., "Toward a Science of Expert Systems," Invited Paper, Computer Science and Statistics: Proceedings of the 18th Symposium on the Interface, American Statistical Association, March, 1986, pgs. 45-52.

5. Heckerman, D.E., "An Axiomatic Framework for Belief Updates," in

Uncertainty in Artificial Intelligence, Vol. 2, J. Lemmer, L. Kanal, ed., North Holland; New York, 1987.

6. * Heckerman, D. E., and Horvitz, E. J., "The Myth of Modularity in Rule-based Systems," in Uncertainty in Artificial Intelligence, Vol. 2, J. Lemmer, L. Kanal, ed., North Holland, New York, 1987.

7. * Heckerman, D.E., and Horvitz, E.J., "On the expressiveness of rule-based systems for reasoning under uncertainty," Proceedings of the National Conference on Artificial Intelligence, Seattle, Washington, July, 1987.

8. * Horvitz, E. J., Heckerman, D. E., Langlotz, C. P., "A framework for comparing alternative formalisms for plausible reasoning," Proceedings of the AAAI," August, 1986, Morgan Kaufman, Los Altos, CA, 1986,

9. * Horvitz, E.J., Breese, J.S., Henrion, M., Decision Theory in Expert Systems and Artificial Intelligence, International Journal of Approximate Reasoning, Elsevier, N.Y. July, 1988.

10. * Heckerman, D.E., An Evaluation of Three Scoring Schemes, Proceedings of the 4th AAAI Workshop on Uncertainty in Artificial Intelligence, Minneapolis, MN., (to appear August 1988).

11. * Horvitz, E.J., A Multiattribute Utility Approach to Inference Understandability and Explanation, Tech. Report, KSL-28-87, Knowledge Systems Laboratory, Stanford, California, March, 1987.

Stanford Project: PROTEAN Project

Principal Investigators:

Oleg Jardetzky
(JARDETZKY@SUMEX-AIM.STANFORD.EDU)
Nuclear Magnetic Resonance Lab, School of Medicine
Stanford University Medical Center
Stanford, California 94305

Bruce G. Buchanan, Ph.D.
(BUCHANAN@SUMEX-AIM.STANFORD.EDU)
Computer Science Department
Stanford University
Stanford, California 94305

The goals of this project are related both to biochemistry and artificial intelligence: (a) use existing AI methods to aid in the determination of the 3-dimensional structure of proteins in solution (not from x-ray crystallography proteins), and (b) use protein structure determination as a test problem for experiments with the AI problem-solving structure known as the Blackboard Model. Empirical data from nuclear magnetic resonance (NMR) and other sources may provide enough constraints on structural descriptions to allow protein chemists to bypass the laborious methods of crystallizing a protein and using X-ray crystallography to determine its structure. This problem exhibits considerable complexity, yet there is reason to believe that AI programs can be written that reason much as experts do to resolve these difficulties. A prototype knowledge-based system assembles major secondary structures of a protein into families of structures compatible with a given set of distance constraints under the control of an explicit assembly strategy. Structures can also be refined at the atomic level of detail using constraints within secondary structures and between amino acid side chains to further restrict the 3-dimensional structure found. By generalizing this approach to the assembly of arrangements of objects subject to constraints, we have developed a language for specifying actions and control for problem solving in similar problem domains.

REFERENCES

1. Altman, R. and Jardetzky, O.: *New strategies for the determination of macromolecular structures in solution.* Journal of Biochemistry (Tokyo), Vol. 100, No. 6, p. 1403-1423, 1986.

2. Altman, R. and Buchanan, B.G.: *Partial Compilation of Control Knowledge.* To appear in: Proceedings of the AAAI 1987.

3. Brinkley, J., Cornelius, C., Altman, R., Hayes-Roth, B. Lichtarge, O., Duncan, B., Buchanan, B.G., Jardetzky, O.: *Application of Constraint Satisfaction Techniques to the Determination of Protein Tertiary Structure.* Report KSL-86-28, Department of Computer Science, 1986.

4. Brinkley, James F., Buchanan, Bruce G., Altman, Russ B., Duncan, Bruce S., Cornelius, Craig W.: *A Heuristic Refinement Method for Spatial Constraint Satisfaction Problems.* Report KSL 87-05, Department of Computer Science.

5. Buchanan, B.G., Hayes-Roth, B., Lichtarge, O., Altman, A., Brinkley, J., Hewett, M., Cornelius, C., Duncan, B., Jardetzky, O.:*The Heuristic Refinement Method for Deriving Solution Structures of Proteins.* Report KSL-85-41. October 1985.

6. Garvey, Alan, Cornelius, Craig, and Hayes-Roth, Barbara: *Computational Costs versus Benefits of Control Reasoning.* Report KSL 87-11, Department of Computer Science.

7. Hayes-Roth, B.: *The Blackboard Architecture: A General Framework for Problem Solving?* Report HPP-83-30, Department of Computer Science, Stanford University, 1983.

8. Hayes-Roth, B.: *BB1: An Environment for Building Blackboard Systems that Control, Explain, and Learn about their own Behavior.* Report HPP-84-16, Department of Computer Science, Stanford University, 1984.

9. Hayes-Roth, B.: *A Blackboard Architecture for Control.* Artificial Intelligence 26:251-321, 1985.

10. Hayes-Roth, B. and Hewett, M.: *Learning Control Heuristics in BB1.* Report HPP-85-2, Department of Computer Science, 1985.

11. Hayes-Roth, B., Buchanan, B.G., Lichtarge, O., Hewett, M., Altman, R., Brinkley, J., Cornelius, C., Duncan, B., and Jardetzky, O.: *PROTEAN: Deriving protein structure from constraints.* Proceedings of the AAAI, 1986, p. 904-909.

12. Jardetzky, O.: *A Method for the Definition of the Solution Structure of Proteins from NMR and Other Physical Measurements: The LAC-Repressor Headpiece.* Proceedings of the International Conference on the Frontiers of Biochemistry and Molecular Biology, Alma Alta, June 17-24, 1984, October, 1984.

13. Lichtarge, Olivier: *Structure determination of proteins in solution by NMR.* Ph.D. Thesis, Stanford University, November, 1986.

14. Lichtarge, Olivier, Cornelius, Craig W., Buchanan, Bruce G., Jardetzky, Oleg: *Validation of the First Step of the Heuristic Refinement Method for the Derivation of Solution Structures of Proteins from NMR Data.*, April 1987. Submitted to Proteins: Structure, Function, and Genetics.

National AIM Project: KNOWLEDGE ENGINEERING FOR RADIATION THERAPY

Principal Investigator:
Ira J. Kalet, Ph.D.
School of Medicine
University of Washington at Seattle
Seattle, Washington 98195
(206) 548-4107

We are developing an expert system for planning of radiation therapy for head and neck cancers. The project will ultimately combine knowledge-based planning with numerical simulation of the radiation treatments. The numerical simulation is needed in order to determine if the proposed treatment will conform to the goals of the plan (required tumor dose, limiting dose to critical organs). The space of possible radiation treatments is numerically very large, making traditional search techniques impractical. Yet, with modern radiation therapy equipment, the design of treatment plans might be significantly aided by automatically generating plans that meet the treatment constraints. The project will result in systematization of knowledge about radiation treatment design, and will also provide an example of how to represent and solve design problems with a knowledge based system.

This project has some relevance to computer science as well, in that our approach, if successful, may contribute to a better understanding of design problem solving with knowledge-based systems.

REFERENCES

1. I. Kalet and W. Paluszynski: A Production Expert System for Radiation Therapy Planning. Proceedings of the AAMSI Congress 1985, May 20-22, 1985, San Francisco, California. Edited by Allan H. Levy and Ben T. Williams. American Association for Medical Systems and Informatics, Washington, D.C., 1985.

2. W. Paluszynski and I. Kalet: Radiation Therapy Planning: A Design Oriented Expert System. WESTEX-87 (Western Conference on Expert Systems), Anaheim, California, June 2-4, 1987.

3. I. Kalet and J. Jacky: Knowledge-based Computer Simulation for Radiation Therapy Planning. Proceedings of the Ninth International Conference on the use of Computers in Radiotherapy, Scheveningen, the Netherlands, June 1987. North Holland, 1987.

4. I. Kalet and W. Paluszynski: Knowledge-based Computer Systems for Radiotherapy Planning. American Journal of Clinical Oncology. In press.

Stanford Project: RADIX -- DERIVING KNOWLEDGE FROM TIME-ORIENTED CLINICAL DATABASES

PENGUIN -- Applying Database and Knowledge base Technology to Medical Instrumentation

Principal Investigators:

Gio C.M. Wiederhold, Ph.D.
Departments of Computer Science
and Medicine
Stanford University
Stanford, California 94305
(415) 497-0685 (WIEDERHOLD@SUMEX-AIM)

Thierry Barsalou, M.D.
Medical Computer Science
Stanford University
(BARSALOU@SUMEX-AIM)

Robert L. Blum, M.D., Ph.D.
Stanford University
(BLUM@SUMEX-AIM)

The RADIX research has been phased out during this year. RADIX had two main goals: (1) to explore the usefulness of knowledge-based techniques to derive medical knowledge from clinical database (DB) systems containing non-randomized, non-protocol patient observations and (2) to develop a program and set of techniques for automated summarization of patient records. The process of reliably deriving causal relationships has proven to be quite difficult because of the complexity of disease states and time relationships, strong sources of bias, and problems of missing and outlying data. However, the experience gained in the RADIX project and its predecessor project, RX, continues to influence basic research directions. The problem of automated knowledge acquisition remains an important area of research, and we expect that the foundations laid here will influence work of others as well.

PENGUIN is a project aimed at developing methods for bringing together expert system and database technologies in an integrated advice system. Databases and expert systems share a common goal -- generating useful information for action -- but accomplish their tasks separately, using different principles. It is clear, however, that future information systems will require both the problem-solving capabilities of expert systems (ESs) and the data-handling capabilities of database management systems (DBMSs). Indeed, combining database and expert system technologies into expert database systems (EDSs) is an emerging research area. One can define an EDS as "a system for developing applications requiring knowledge-directed processing of shared information". From a perspective of developing advanced biomedical information systems, this definition conveys two goals: (1) enhancing DBMSs with structuring and manipulation tools that take more semantics into account; (2) allowing ESs to access and to handle efficiently information stored in database(s). In this research project, we investigate the hypothesis that the object-oriented approach to knowledge and information representation can serve as a unifying scheme for developing EDSs.

We explore this hypothesis in a practical biomedical environment, Fluorescence Activated Cell Sorting (FACS). FACS is emerging as a major source of information for biomedical research and clinical practice. Currently, achieving good FACS performance requires analyzing and integrating various complex data and knowledge sources in order to fulfill the information needs of FACS investigators. This work involves a close collaboration between researchers in the Medical Information Sciences Program and the Departments of Computer Science and Genetics.

REFERENCES

**PENGUIN PUBLICATIONS**

1. Barsalou, Thierry and Gio Wiederhold: "Automating a Cell Counter"; to be published in The International Journal of Artificial Intelligence in Engineering, Computational Mechanics Publ., UK, 1988.
2. Barsalou, Thierry and Gio Wiederhold: "Applying a Semantic Model to an Immunology Database"; in W.W. Stead (editor), Proceedings of the Eleventh Symposium on Computer Applications in Medical Care, pages 871-877, IEEE Computer Society Press, Washington, D.C., November 1987.
3. Barsalou, Thierry, W.A. Moore, L.A. Herzenberg, and G. Wiederhold: "A Database System to Facilitate the Design of FACS Experiment Protocols" (abstract); Cytometry, Vol.97, August 1987.
4. Barsalou, T. and G. Wiederhold. *Knowledge-based mapping of relations into objects.* To appear in the International Journal of AI in Engineering, 1988.
5. Barsalou, T. *An object-based architecture for biomedical expert database systems.* Submitted to the IEEE Twelfth Symposium on Computer Applications in Medical Care, 1988.

**RADIX PUBLICATIONS**

*Monograph*

1. Blum, R.L.: *Discovery and representation of causal relationships from a large time-oriented clinical database: The RX project.* in D.A.B. Lindberg and P.L. Reichertz (Eds.), LECTURE NOTES IN MEDICAL INFORMATICS, Vol. 19, Springer-Verlag, New York, 1982.

*Journal Articles*

1. Blum, R.L.: *Computer-Assisted Design of Studies using Routine Clinical Data: Analyzing the Association of Prednisone and Cholesterol.* Annals of Internal Medicine 104(6):858-868, June, 1986.
2. Blum, R.L.: *Discovery, confirmation, and incorporation of causal relationships from a large time-oriented clinical database: The RX Project.* Computers and Biomedical Research 15(2):164-187, April, 1982.
3. Walker, M.G. *How Feasible is Automated Discovery?* IEEE Expert 2(1):70-82, Spring, 1987.

*Conference Proceedings*

1. DeZegher-Geets, I.M., Freeman, A.G., Walker, M.G., Blum, R.L., and Wiederhold, G.C.M. *Computer-aided Summarization of a Time-oriented Medical Data Base.* In Proceedings of the Third Annual Conference on Computerization of Medical Records. Institute for Medical Record Economics, 1987.

2. Downs, S., Walker, M.G., and Blum, R.L. *Automated Summarization of On-line Medical Records*. In Proceedings of the Fifth World Congress on Medical Informatics (Medinfo), pages 800-804. Elsevier Science Publishers, 1986.

3. Walker, M.G., and Blum, R.L. *Towards Automated Discovery from Clinical Databases: the RADIX Project*. In Proceedings of the Fifth World Congress on Medical Informatics (Medinfo), pages 32-36. Elsevier Science Publishers, 1986.

Stanford Project: REFEREE Project

Principal Investigators:

Bruce G. Buchanan, Principal Investigator
Computer Science Department
Stanford University
Stanford, California 94305

Byron W. Brown, Co-Principal Investigator
Department of Medicine
Stanford University Medical Center

Daniel E. Feldman, Associate Investigator
Department of Medicine
Stanford University Medical Center

The goals of this project are related both to medical science and artificial intelligence: (a) use AI methods to allow the informed but non-expert reader of the medical literature to evaluate a randomized clinical trial, and (b) use the interpretation of the medical literature as a test problem for studies of knowledge acquisition and fusion of information from disparate sources. REFEREE and REVIEWER, a planned extension, will be used to evaluate the medical literature of clinical trials to determine the quality of a clinical trial, make judgments on the efficacy of the treatment proposed, and synthesize rules of clinical practice. The research is an initial step toward a more general goal - building computer systems to help the clinician and medical scientist read the medical literature more critically and more rapidly for use in making clinical decisions.

REFERENCES

1. Haggerty, J.: *REFEREE and RULECRITIC: Two prototypes for assessing the quality of a medical paper.* REPORT KSL-84-49. Master's Thesis, Stanford University, May 1984.

2. *Chavez, R. Martin and Cooper, G. F.: *KNET: Integration Hypermedia and Normative Bayesian Modeling.* REPORT KSL. Stanford University, March 1988.

3. *Lehmann, H. *Knowledge Acquisition for Probabilistic Expert Systems.* Submitted to Symposium on Computer Applications in Medical Care, 1988.

## References

**1.** Coulter, C. L. "Research Instrument Sharing." *Science 201*, 4354 (1978).

**2.** Lederberg, J. "Digital Communications and the Conduct of Science: The New Literacy." *Proceedings of the IEEE 66*, 11 (1978).

**3.** NLM Planning Panel 4. NLM Long Range Plan: Medical Informatics. Shortliffe, E.H., Panel 4 Chairman , National Library of Medicine, January, 1987.

**4.** Scheifler, R.W. and Gettys, J. The X Window System. Draft October 1986. To appear in a special issue of the *ACM Transactions on Graphics* -- user interface software

**5.** Steele, Guy L., Jr.. *Common Lisp - The Language.* Digital Press, Burlington, MA, 1984.